

# Appendix - LTA with MplusAutomation

---

## Rationale for MplusAutomation workflow:

This R-script is intended to provide a template for running LTA and associated tasks in a systematic manner. Using this approach all code for data pre-processing, model estimation, tabulation, and figure processing can be contained within a single script providing clear documentation. It is the authors belief that this dramatically reduces risk of user-error, is conducive to open-science philosophy, and scientific transparency. All models are estimated using **Mplus** (Muthén & Muthén, 1998 - 2017) using the wrapping program **MplusAutomation** (Hallquist & Wiley, 2018). This method requires that the user to have the proprietary software **Mplus** installed on their OS.

This approach also relies on the utility of **R-Projects**. This provides a structured framework for organizing all associated data files, Mplus text files, scripts, and figures. Given the high output of Mplus files inherent to LTA modeling, creating a system of project sub-folders greatly improves organization (i.e., folders; ‘data’, ‘mplus\_files’ ‘figures’, etc.) Additionally, the communication between R and Mplus requires the specification of file-paths a procedure which is streamlined by use of **R-projects**. Due to the reliance on file-paths the **here** package is utilized for reproducibility, by making all path syntax uniform across operating systems.

---

**Data Source:** The data used to illustrate these analyses include elementary school student *Science Attitude* survey items collected during 7th and 10th grades from the **Longitudinal Study of American Youth** (LSAY; Miller, 2015).

---

## Preparation

### Download the R-Project

Download Github repository here: <https://github.com/garberadamc/LTA-FAQ>

For readers **unfamiliar with Github** and version controlled R-projects:

1. On the repository page, click the green **Code** button and in the menu choose option **Download ZIP**
2. Place the un-zipped downloaded folder on your desktop
3. Within this folder open the file with the blue cube icon that is file type **LTA-FAQ-analyses.Rproj**
4. Next open the file containing all analysis code named **LTA-FAQ-analyses.R**.

**Note:** Alternatively, if preferred users may follow analyses using the Rmarkdown script (**.Rmd**).

## Project folder organization

The following sub-folders will be used to contain files:

1. "data"
2. "enum\_LCA\_time1"
3. "enum\_LCA\_time2"
4. "LTA\_models"
5. "figures"

### Note regarding choice of project location:

If the main project folder is located within too many nested folders it may result in a file-path error when estimating models with `MplusAutomation`.

---

## Notation guide

In the following script, three types of comments are included in code blocks in which models are estimated using `MplusAutomation`.

- a. **Annotate in R:** The hashtag symbol `#` identifies comments written in R-language form.
  - b. **Annotate in Mplus input:** Within the `mplusObject()` function all text used to generate Mplus input files is enclosed within quotation marks (green text). To add comments the Mplus language convention is used (e.g., `!!! annotate Mplus input !!!`).
  - c. **Annotate context-specific syntax:** To signal to the user areas of the syntax which vary based on the particular modeling context the text, `NOTE CHANGE:` is used.
- 

To install package `{rhdf5}`

```
if (!requireNamespace("BiocManager", quietly = TRUE))  
  install.packages("BiocManager")  
  
BiocManager::install("rhdf5")
```

Load packages

```
library(MplusAutomation)  
library(rhdf5)  
library(tidyverse)  
library(here)  
library(glue)  
library(janitor)  
library(gt)  
library(reshape2)  
library(cowplot)
```

Read in LSAY data file

*Note:* The LSAY data file, `lsay_lta_faq_2020.csv`, has been pre-processed.

```
lsay_data <- read_csv(here("data", "lsay_lta_faq_2020.csv"),
                      na=c("9999", "9999.00"))
```

---

## Step 1: Enumeration

---

### Enumerate time point 1 (7th grade)

```
# NOTE CHANGE: '1:6' indicates the number of k-class models to estimate.
# User can change this number to fit research context.
# In this example, the code loops or iterates over values 1 through 6 ( '{k}' ).
t1_enum_k_16 <- lapply(1:6, function(k) {
  enum_t1 <- mplusObject(

# The 'glue' function inserts R code within a string or "quoted green text" using the syntax {---}
  TITLE = glue("Class-{k}_Time1"),

  VARIABLE = glue(
    "!!! NOTE CHANGE: List of the five 7th grade science attitude indicators !!!
    categorical = ab39m-ab39x;
    usevar = ab39m-ab39x;

    classes = c({k});"),

  ANALYSIS =
    "estimator = mlr;
    type = mixture;
    !!! NOTE CHANGE: The intial and final start values. Reduce to speed up estimation time. !!!
    starts = 500 100;
    processors=10;",

  OUTPUT = "sampstat residual tech11 tech14;",

  PLOT =
    "type = plot3;
    series = ab39m-ab39x(*);",

  usevariables = colnames(lsay_data),
  rdata = lsay_data)

# NOTE CHANGE: Fix to match appropriate sub-folder name
# See after `here` function (e.g., "enum_LCA_time1")
enum_t1_fit <- mplusModeler(enum_t1,
```

```

        dataout=here("enum_LCA_time1", "t1.dat"),
        modelout=glue(here("enum_LCA_time1", "c{k}_lca_enum_time1.inp")),
        check=TRUE, run = TRUE, hashfilename = FALSE)
})

```

**NOTE:** Even after successfully estimating a series of models the `mplusModeler()` function will return error messages.

---

## Enumerate time point 2 (10th grade)

```

t2_enum_k_16 <- lapply(1:6, function(k) {
  enum_t2 <- mplusObject(

    TITLE = glue("Class-{k}_Time2"),

    VARIABLE =
    glue(
      "!!! NOTE CHANGE: List of the five 10th grade science attitude indicators !!!
      categorical = ga33a-ga33l;
      usevar = ga33a-ga33l;

      classes = c({k});"),

    ANALYSIS =
      "estimator = mlr;
      type = mixture;
      starts = 500 100;
      processors=10;",

    OUTPUT = "sampstat residual tech11 tech14;",

    PLOT =
      "type = plot3;
      series = ga33a-ga33l(*);",

    usevariables = colnames(lsay_data),
    rdata = lsay_data)

  enum_t2_fit <- mplusModeler(enum_t2,
    dataout=here("enum_LCA_time2", "t2.dat"),
    modelout=glue(here("enum_LCA_time2", "c{k}_lca_enum_time2.inp")),
    check=TRUE, run = TRUE, hashfilename = FALSE)
})

```

---

## Compare time 1 & time 2 LCA plots

---

Read models

```
# timepoint 1
output_enum_t1 <- readModels(here("enum_LCA_time1"), quiet = TRUE)
# timepoint 2
output_enum_t2 <- readModels(here("enum_LCA_time2"), quiet = TRUE)
```

### Plot time 1 LCA

Extract posterior probabilities

```
# NOTE CHANGE (below): chosen model (i.e., 'c4') is based on enumeration decision
plot_t1 <- as.data.frame(output_enum_t1
  [["c4_lca_enum_time1.out"]]
  [["gh5"]][["means_and_variances_data"]]
  [["estimated_probs"]][["values"]]
  [seq(2, 10, 2),])
# NOTE CHANGE (above): value '10' is determined by 2 times the number of items in LCA model
```

Extract class size proportions

```
c_size <- as.data.frame(output_enum_t1
  [["c4_lca_enum_time1.out"]]
  [["class_counts"]][["modelEstimated"]][["proportion"]])

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

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

Rename class and indicator names

```
colnames(plot_t1) <- paste0("C", 1:4, glue(" ({c_size[1:4,]}%))")

plot_t1 <- cbind(Var = paste0("U", 1:5), plot_t1)

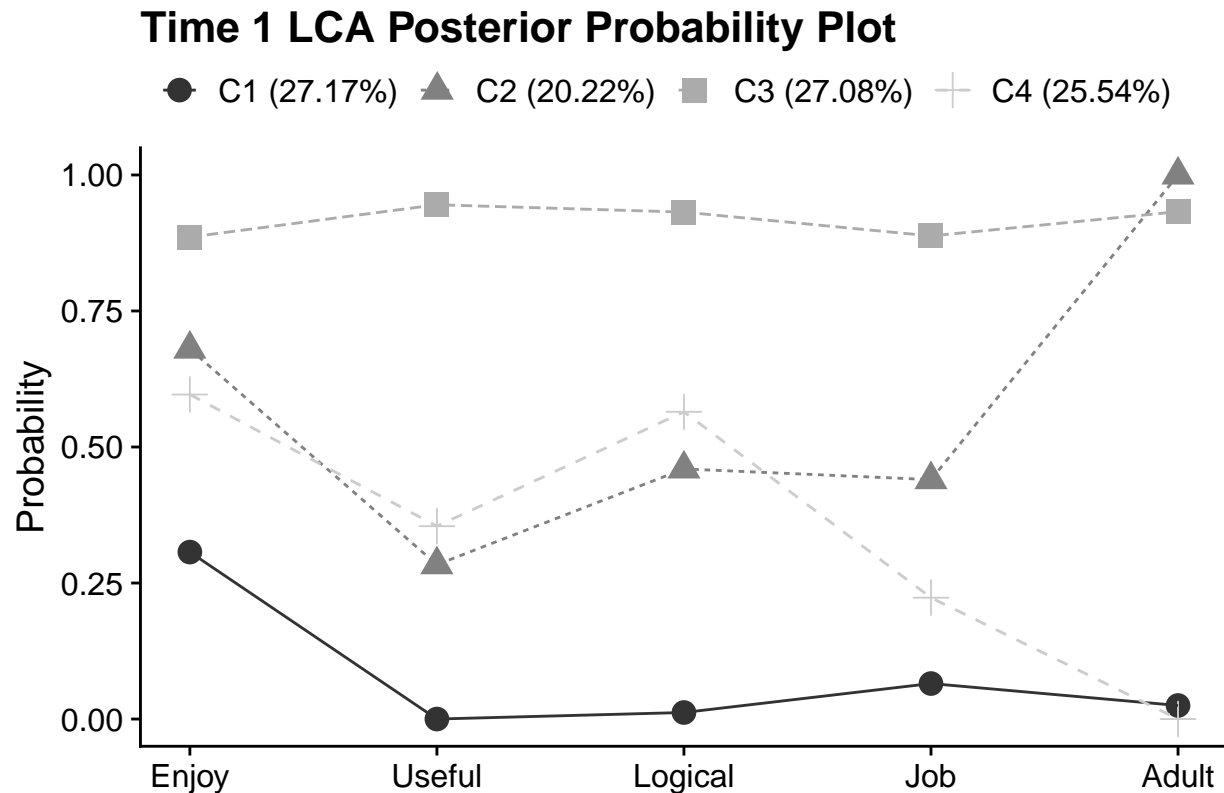
plot_t1$Var <- factor(plot_t1$Var,
  levels = c("U1", "U2", "U3", "U4", "U5"), # NOTE CHANGE: the number of levels
  labels = c("Enjoy", "Useful", "Logical", "Job", "Adult"))
```

Change data frame from wide to long format

```
pd_long_t1 <- melt(plot_t1, id.vars = "Var")
```

Plot posterior probabilities (7th grade)

```
pd_long_t1 %>%
  ggplot(aes(x = as.integer(Var), y = value,
             shape = variable, colour = variable, lty = variable)) +
  geom_point(size = 4) + geom_line() +
  scale_x_continuous("", breaks = 1:5, labels = plot_t1$Var) +
  scale_colour_grey() +
  labs(title = "Time 1 LCA Posterior Probability Plot", y = "Probability") +
  theme_cowplot() +
  theme(legend.title = element_blank(),
        legend.position = "top")
```



Save plot to the figures folder

```
ggsave(here("figures", "T1_C4_LCA_plot.png"), dpi=300, height=5, width=7, units="in")
```

### Plot time 2 LCA

```
# extract posterior probabilities
plot_t2 <- as.data.frame(output_enum_t2
  [["c4_lca_enum_time2.out"]]
  [["gh5"]][["means_and_variances_data"]]
  [["estimated_probs"]][["values"]]
  [seq(2, 10, 2),]) #seq("from", "to", "by")
```

Extract class size proportions

```

c_size <- as.data.frame(output_enum_t2[["c4_lca_enum_time2.out"]]
                        [["class_counts"]][["modelEstimated"]][["proportion"]])

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

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

```

Rename class and indicator names

```

colnames(plot_t2) <- paste0("C", 1:4, glue(" ({c_size[1:4,]}%)"))
plot_t2 <- cbind(Var = paste0("U", 1:5), plot_t1)

plot_t2$Var <- factor(plot_t2$Var,
                      levels = c("U1", "U2", "U3", "U4", "U5"),
                      labels = c("Enjoy", "Useful", "Logical", "Job", "Adult"))

```

Change data frame from wide to long format

```

pd_long_t2 <- melt(plot_t2, id.vars = "Var")

```

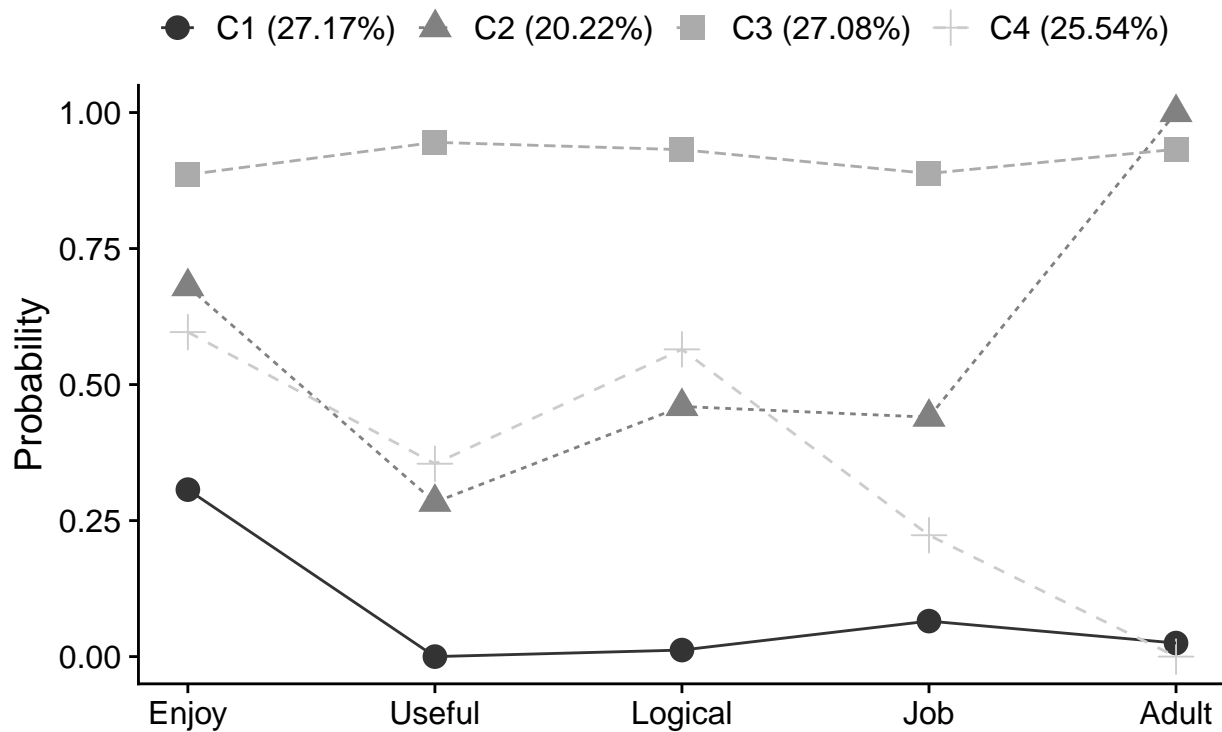
Plot posterior probabilities (10th grade)

```

pd_long_t2 %>%
  ggplot(aes(x = as.integer(Var), y = value,
             shape = variable, colour = variable, lty = variable)) +
  geom_point(size = 4) +
  geom_line() +
  scale_x_continuous("", breaks = 1:5, labels = plot_t2$Var) +
  scale_colour_grey() +
  labs(title = "Time 2 LCA Posterior Probability Plot", y = "Probability") +
  theme_cowplot() +
  theme(legend.title = element_blank(),
        legend.position = "top")

```

## Time 2 LCA Posterior Probability Plot



Save plot to the figures folder

```
ggsave(here("figures", "T2_C4_LCA_plot.png"), dpi=300, height=5, width=7, units="in")
```

## Step 2: Create model fit summary table

Extract model fit data

```
enum_extract1 <- LatexSummaryTable(output_enum_t1,
  keepCols=c("Title", "Parameters", "LL", "BIC", "aBIC",
    "BLRT_PValue", "T11_VLMR_PValue", "Observations"))

enum_extract2 <- LatexSummaryTable(output_enum_t2,
  keepCols=c("Title", "Parameters", "LL", "BIC", "aBIC",
    "BLRT_PValue", "T11_VLMR_PValue", "Observations"))
```

Calculate indices derived from the Log Likelihood (LL)



```

allFit1 <- enum_extract1 %>%
  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)) %>%
  select(1:5,9:10,6:7,13,14) %>%
  arrange(Parameters)

allFit2 <- enum_extract2 %>%
  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)) %>%
  select(1:5,9:10,6:7,13,14) %>%
  arrange(Parameters)

allFit <- full_join(allFit1,allFit2)

```

---

Format table

```

allFit %>%
  mutate(Title = str_remove(Title, "_Time*")) %>%
  gt() %>%
  tab_header(
    title = md("**Model Fit Summary Table**"), subtitle = md("&nbsp;") %>%
    tab_source_note(
      source_note = md("Data Source: **Longitudinal Study of American Youth.**") %>%
  cols_label(
    Title = "Classes",
    Parameters = md("Par"),
    LL = md("*LL*"),
    T11_VLMR_PValue = "VLMR",
    BLRT_PValue = "BLRT",
    BF = md("BF"),
    cmPk = md("*cmP_k*") %>%
  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(10, decimals = 2,
            drop_trailing_zeros=TRUE,
            suffixing = TRUE) %>%
fmt_number(c(3:9,11), decimals = 2) %>%
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))) %>%
tab_row_group(
  group = "Time-1",
  rows = 1:6) %>%
tab_row_group(
  group = "Time-2",
  rows = 7:12) %>%
row_group_order(
  groups = c("Time-1", "Time-2"))

```

Model Fit Summary Table<sup>1</sup>

Classes	Par	LL	BIC	aBIC	CIAC	AWE	BLRT	VLMR	BF	cmP_k
Time-1										
C1_LCA1	5	-10,250.60	20,541.34	20,525.45	20,546.34	20,596.47	–	–	–	<.001
C2_LCA1	11	-8,785.32	17,658.92	17,623.97	17,669.93	17,780.22	<.001	<.001	>100	<.001
C3_LCA1	17	-8,693.57	17,523.59	17,469.57	17,540.59	17,711.04	<.001	<.001	>100	0.00
C4_LCA1	23	-8,664.09	17,512.79	17,439.71	17,535.79	17,766.40	<.001	<.001	>100	1.00
C5_LCA1	29	-8,662.39	17,557.54	17,465.39	17,586.54	17,877.31	1.00	0.66	>100	<.001
C6_LCA1	35	-8,661.54	17,604.01	17,492.80	17,639.01	17,989.94	1.00	0.75	>100	<.001
Time-2										
C1_LCA2	5	-7,658.79	15,356.19	15,340.30	15,361.19	15,409.80	–	–	–	<.001
C2_LCA2	11	-6,073.81	12,232.56	12,197.61	12,243.56	12,350.50	<.001	<.001	>100	<.001
C3_LCA2	17	-5,988.36	12,107.99	12,053.98	12,124.99	12,290.27	<.001	<.001	>100	0.32
C4_LCA2	23	-5,964.45	12,106.50	12,033.43	12,129.51	12,353.12	<.001	0.00	2.1	0.68
C5_LCA2	29	-5,961.68	12,147.30	12,055.16	12,176.30	12,458.25	0.31	0.36	>100	<.001
C6_LCA2	35	-5,961.26	12,192.79	12,081.59	12,227.79	12,568.07	1.00	0.50	>100	<.001

<sup>1</sup>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.

Data Source: **Longitudinal Study of American Youth.**

---

## Step 3: Estimate Latent Transition Analysis

---

### Estimate non-invariant estimated LTA model

---

```
lta_non_inv <- mplusObject(

  TITLE =
    "4-Class-Non-Invariant",

  VARIABLE =
    "usev = ab39m ab39t ab39u ab39w ab39x ! 7th grade indicators
      ga33a ga33h ga33i ga33k ga33l; ! 10th grade indicators

    categorical = ab39m-ab39x ga33a-ga33l;

    classes = c1(4) c2(4);",

  ANALYSIS =
    "estimator = mlr;
    type = mixture;
    starts = 500 100;
    processors=10;",

  MODEL =
    "%overall%
    c2 on c1; !!! estimate all multinomial logistic regressions !!!

    !!! The above syntax can also be written as: !!!
      ! c2#1 on c1#1 c1#2 c1#3; !
      ! c2#2 on c1#1 c1#2 c1#3; !
      ! c2#3 on c1#1 c1#2 c1#3; !

    MODEL c1: !!! the following syntax will estimate item thresholds freely !!!
    %c1#1%
    [AB39M$1-AB39X$1];
    %c1#2%
    [AB39M$1-AB39X$1];
    %c1#3%
    [AB39M$1-AB39X$1];
    %c1#4%
    [AB39M$1-AB39X$1];

    MODEL c2:
    %c2#1%
```

```

[GA33A$1-GA33L$1];
%c2#2%
[GA33A$1-GA33L$1];
%c2#3%
[GA33A$1-GA33L$1];
%c2#4%
[GA33A$1-GA33L$1];",

OUTPUT = "tech1 tech15 svalues;",

usevariables = colnames(lsay_data),
rdata = lsay_data)

lta_non_inv_fit <- mplusModeler(lta_non_inv,
                              dataout=here("LTA_models", "lta.dat"),
                              modelout=here("LTA_models", "4-class-non-invariant.inp"),
                              check=TRUE, run = TRUE, hashfilename = FALSE)

```

---

## Estimate invariant LTA model

---

```

lta_inv <- mplusObject(

  TITLE =
    "4-Class-Invariant",

  VARIABLE =
    "usev = ab39m ab39t ab39u ab39w ab39x ! 7th grade indicators
      ga33a ga33h ga33i ga33k ga33l; ! 10th grade indicators

    categorical = ab39m-ab39x ga33a-ga33l;

    classes = c1(4) c2(4);",

  ANALYSIS =
    "estimator = mlr;
    type = mixture;
    starts = 500 100;
    processors=10;",

  MODEL =
    "%overall%
    c2 on c1;

    MODEL c1: !!! the following syntax will estimate item thresholds constrained across latent variables !!!
    %c1#1%
    [AB39M$1-AB39X$1] (1-5); !!! labels that are repeated will constrain parameters equality !!!

```

```

%c1#2%
[AB39M$1-AB39X$1] (6-10);
%c1#3%
[AB39M$1-AB39X$1] (11-15);
%c1#4%
[AB39M$1-AB39X$1] (16-20);

MODEL c2:
%c2#1%
[GA33A$1-GA33L$1] (1-5);
%c2#2%
[GA33A$1-GA33L$1] (6-10);
%c2#3%
[GA33A$1-GA33L$1] (11-15);
%c2#4%
[GA33A$1-GA33L$1] (16-20);",

SAVEDATA =
"file = LTA_Inv_CPROBS.dat;
save = cprob;
missflag = 9999;",

OUTPUT = "tech1 tech15 svalues;",

usevariables = colnames(lsay_data),
rdata = lsay_data)

lta_inv_fit <- mplusModeler(lta_inv,
  dataout=here("LTA_models", "lta.dat"),
  modelout=here("LTA_models", "4-class-invariant.inp"),
  check=TRUE, run = TRUE, hashfilename = FALSE)

```

Conduct Satorra-Bentler  $\chi^2$  difference testing

- non-invariant (comparison): This model has **more** parameters.
- invariant (nested): This model has **less** parameters.

```
# *0 = null or nested model & *1 = comparison or parent model
```

```
lta_models <- readModels(here("LTA_models"), quiet = TRUE)
```

```

T0 <- lta_models[["X4.class.invariant.out"]][["summaries"]][["ChiSqCategoricalLRT_Value"]] # test
T1 <- lta_models[["X4.class.non.invariant.out"]][["summaries"]][["ChiSqCategoricalLRT_Value"]] # test
c0 <- lta_models[["X4.class.invariant.out"]][["summaries"]][["LLCorrectionFactor"]] # scaling
c1 <- lta_models[["X4.class.non.invariant.out"]][["summaries"]][["LLCorrectionFactor"]] # scaling
d0 <- lta_models[["X4.class.invariant.out"]][["summaries"]][["ChiSqCategoricalLRT_DF"]] # degree
d1 <- lta_models[["X4.class.non.invariant.out"]][["summaries"]][["ChiSqCategoricalLRT_DF"]] # degree
df <- abs(d0-d1)

```

```
# Satorra-Bentler scaled Difference test equations
```

```

cd <- (((d0*c0)-(d1*c1))/(d0-d1))
t <- (((T0*c0)-(T1*c1))/(cd))

```

```
t      # Satorra-Bentler chi-square value
df     # difference test degrees of freedom

# Significance test
(p_diff <- pchisq(t, df, lower.tail=FALSE))
```

**RESULT:** The Satorra-Bentler scaled  $\chi^2$  difference test comparing the invariant and non-invariant LTA models was,  $T(19) = 20.05, p = .39$ .

---

Alternate, less verbose way to run LTA with the `createMixtures` function.

```
data <- lsay_data %>% select(5:14) # select only the indicator variables

createMixtures(
  classes = 4,
  filename_stem = "sci_attitude",
  model_overall = "c2 ON c1;",
  model_class_specific = c(
    "[ab39m$1] (a{C}); [ab39t$1] (b{C}); [ab39u$1] (c{C}); [ab39w$1] (d{C}); [ab39x$1] (e{C});",
    "[ga33a$1] (a{C}); [ga33h$1] (b{C}); [ga33i$1] (c{C}); [ga33k$1] (d{C}); [ga33l$1] (e{C});"),
  rdata = data,
  ANALYSIS = "PROCESSORS IS 10; STARTS = 500 100; PARAMETERIZATION = PROBABILITY;",
  VARIABLE = "CATEGORICAL = ab39m-ab39x ga33a-ga33l;")

runModels(filefilter = "sci_attitude")

results <- readModels(filefilter = "sci_attitude")
```

Read invariance model and extract parameters (intercepts and multinomial regression coefficients)

```
lta_inv1 <- readModels(here("LTA_models", "4-Class-Invariant.out" ), quiet = TRUE)

par <- as_tibble(lta_inv1[["parameters"]][["unstandardized"]]) %>%
  select(1:3) %>%
  filter(grepl('ON|Means', paramHeader)) %>%
  mutate(est = as.numeric(est))
```

Manual method to calculate transition probabilities:

Although possible to extract transition probabilities directly from the output the following code illustrates how the parameters are used to calculate each transition. This is useful for conducting advanced LTA model specifications such as making specific constraints within or between transition matrices.

```
# Name each parameter individually to make the subsequent calculations more readable
a1 <- unlist(par[13,3]); a2 <- unlist(par[14,3]); a3 <- unlist(par[15,3]); b11 <- unlist(par[1,3]);
b21 <- unlist(par[4,3]); b31 <- unlist(par[7,3]); b12 <- unlist(par[2,3]); b22 <- unlist(par[5,3]);
b32 <- unlist(par[8,3]); b13 <- unlist(par[3,3]); b23 <- unlist(par[6,3]); b33 <- unlist(par[9,3])

# Calculate transition probabilities from the logit parameters
```

```

t11 <- exp(a1+b11)/(exp(a1+b11)+exp(a2+b21)+exp(a3+b31)+exp(0))
t12 <- exp(a2+b21)/(exp(a1+b11)+exp(a2+b21)+exp(a3+b31)+exp(0))
t13 <- exp(a3+b31)/(exp(a1+b11)+exp(a2+b21)+exp(a3+b31)+exp(0))
t14 <- 1 - (t11 + t12 + t13)

t21 <- exp(a1+b12)/(exp(a1+b12)+exp(a2+b22)+exp(a3+b32)+exp(0))
t22 <- exp(a2+b22)/(exp(a1+b12)+exp(a2+b22)+exp(a3+b32)+exp(0))
t23 <- exp(a3+b32)/(exp(a1+b12)+exp(a2+b22)+exp(a3+b32)+exp(0))
t24 <- 1 - (t21 + t22 + t23)

t31 <- exp(a1+b13)/(exp(a1+b13)+exp(a2+b23)+exp(a3+b33)+exp(0))
t32 <- exp(a2+b23)/(exp(a1+b13)+exp(a2+b23)+exp(a3+b33)+exp(0))
t33 <- exp(a3+b33)/(exp(a1+b13)+exp(a2+b23)+exp(a3+b33)+exp(0))
t34 <- 1 - (t31 + t32 + t33)

t41 <- exp(a1)/(exp(a1)+exp(a2)+exp(a3)+exp(0))
t42 <- exp(a2)/(exp(a1)+exp(a2)+exp(a3)+exp(0))
t43 <- exp(a3)/(exp(a1)+exp(a2)+exp(a3)+exp(0))
t44 <- 1 - (t41 + t42 + t43)

```

Create transition table

```

t_matrix <- tibble(
  "Time1" = c("C1=1", "C1=2", "C1=3", "C1=4"),
  "C2=1" = c(t11, t21, t31, t41),
  "C2=2" = c(t12, t22, t32, t42),
  "C2=3" = c(t13, t23, t33, t43),
  "C2=4" = c(t14, t24, t34, t44))

t_matrix %>%
  gt(rowname_col = "Time1") %>%
  tab_stubhead(label = "7th grade") %>%
  tab_header(
    title = md("**Student transitions from 7th grade (rows) to 10th grade (columns)**"),
    subtitle = md("&nbsp;") %>%
  )
fmt_number(2:5, decimals = 2) %>%
  tab_spanner(label = "10th grade", columns = 2:5)

```

Student transitions from 7th grade (rows) to 10th grade (columns)

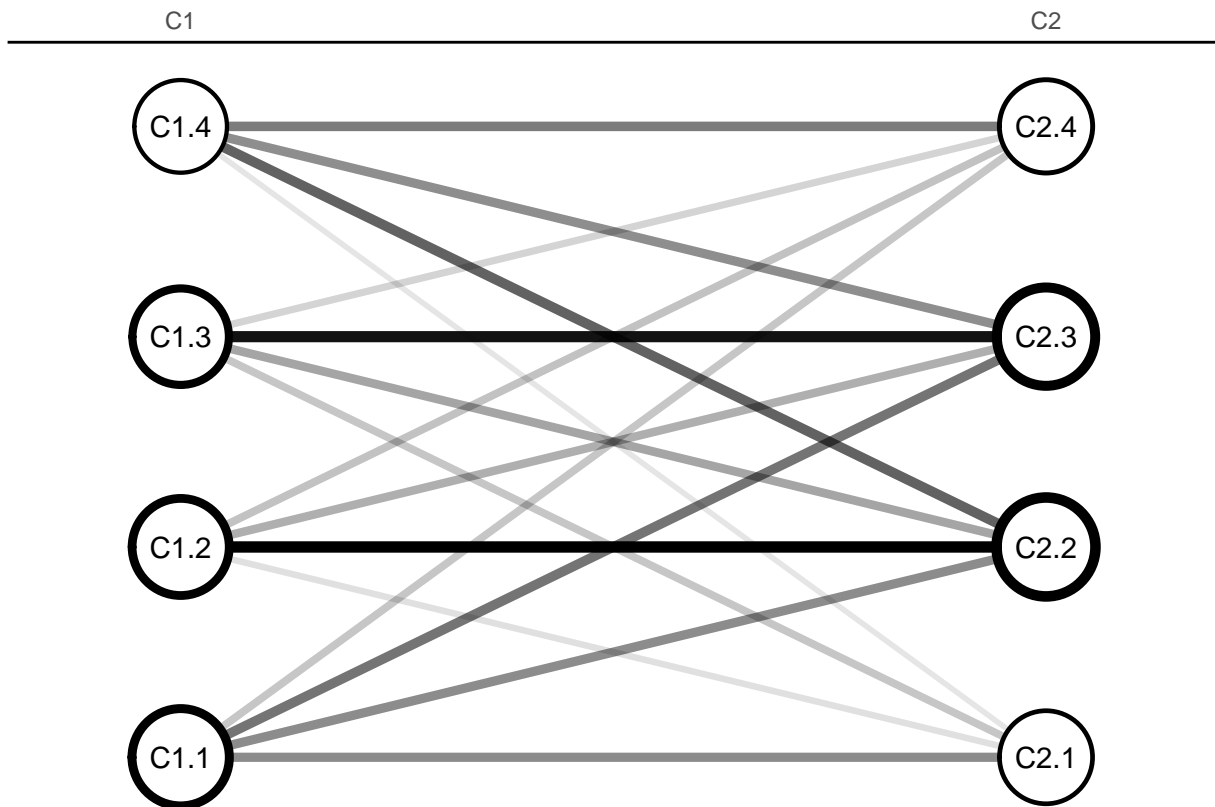
7th grade	10th grade			
	C2=1	C2=2	C2=3	C2=4
C1=1	0.27	0.27	0.32	0.15
C1=2	0.09	0.56	0.19	0.16
C1=3	0.15	0.21	0.52	0.12
C1=4	0.08	0.35	0.27	0.30

---

## Plot LTA transitions

---

```
MplusAutomation::plotLTA(lta_inv1)
```



---

## References

- Hallquist, Michael N., and Joshua F. Wiley. 2018. "MplusAutomation: An R Package for Facilitating Large-Scale Latent Variable Analyses in Mplus." *Structural Equation Modeling*, 1–18. <https://doi.org/10.1080/10705511.2017.1402334>.
- Miller, Jon D. Longitudinal Study of American Youth (LSAY), Seventh Grade Data, 1987-1988; 2015-2016. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor], 2019-04-23. <https://doi.org/10.3886/ICPSR37287.v1>
- Müller, Kirill. 2017. Here: A Simpler Way to Find Your Files. <https://CRAN.R-project.org/package=here>.
- Muthén L.K., & Muthén B.O. (1998-2017) Mplus User's Guide. Eight Edition. Los Angeles, CA: Muthén & Muthén.



R Core Team. 2019. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2020. Dplyr: A Grammar of Data Manipulation. <https://CRAN.R-project.org/package=dplyr>.

Wickham, Hadley, Jim Hester, and Winston Chang. 2020. Devtools: Tools to Make Developing R Packages Easier. <https://CRAN.R-project.org/package=devtools>.

Wickham, Hadley, Jim Hester, and Romain François. 2018. Readr: Read Rectangular Text Data. <https://CRAN.R-project.org/package=readr>.1

---