

Supplemental Materials B - Implement LTA Analyses with MplusAutomation

Rationale for MplusAutomation workflow:

This R-script is intended to provide a template for running LTA and RI-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 our 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 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 number of Mplus files that are associated with running LTA models, 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

Link to 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.Rproj**

4. Next open the file containing all analysis code named `LTA-FAQ-analyses.R`.

Note: Users may also follow these 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 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 within quotations 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(gt)  
library(reshape2)  
library(cowplot)  
library(patchwork)
```

Read in the LSAY data file named `lsay_lta_faq_2021.csv`.

```
lsay_data <- read_csv(her("data", "lsay_lta_faq_2021.csv"),
                      na=c("9999", "9999.00"))
```

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: It is highly recommended that you check the Mplus output files (.out) to check for convergence warnings or syntax errors. Mplus files may be viewed in the RStudio window (bottom right pane).

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)
})
```

Create Model Fit Summary Table

The following syntax for producing publication ready fit tables can be cited using the following citation:

Garber, A. C. (2021). Creating Summary Fit Tables for LCA and LTA Analyses Using MplusAutomation. Retrieved from psyarxiv.com/uq2fh

Read all models for enumeration table

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

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 Fit Table

```
allFit %>%
  mutate(Title = str_remove(Title, "_Time*")) %>%
  gt() %>%
  tab_header(
    title = md("**Model Fit Summary Table**"), subtitle = md("&nbsp;")) %>%
  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(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 Table¹

Classes	Par	LL	BIC	aBIC	CIAC	AWE	BLRT	VLMR	BF	cmPk
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

¹Note. 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.

Compare Time 1 & Time 2 Condition Item Probability Plots

Read models for plotting (4-class models)

```
model_t1_c4 <- readModels(here("enum_LCA_time1", "c4_lca_enum_time1.out"), quiet = TRUE)
model_t2_c4 <- readModels(here("enum_LCA_time2", "c4_lca_enum_time2.out"), quiet = TRUE)
```

Create a function `plot_lca_function` that requires 5 arguments:

- `model_name`: name of Mplus model object (e.g., `model_t1_c4`)
- `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., 4)
- `item_labels`: the item labels for x-axis (e.g., `c("Enjoy", "Useful", "Logical", "Job", "Adult")`)
- `plot_title`: include the title of the plot here (e.g., "Time 1 LCA Conditional Item Probability Plot")

```

plot_lca_function <- function(model_name,item_num,class_num,item_labels,plot_title){

mplus_model <- as.data.frame(model_name$gh5$means_and_variances_data$estimated_probs$values)
plot_t1 <- 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_t1) <- paste0("C", 1:class_num, glue(" ({c_size[1:class_num,]}%)"))

plot_t1 <- cbind(Var = paste0("U", 1:item_num), plot_t1)
plot_t1$Var <- factor(plot_t1$Var,
                      labels = item_labels)
plot_t1$Var <- fct_inorder(plot_t1$Var)
pd_long_t1 <- melt(plot_t1, id.vars = "Var")

p <- 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 = plot_title, y = "Probability") +
  theme_cowplot() +
  theme(legend.title = element_blank(),
        legend.position = "top")

p
return(p)
}

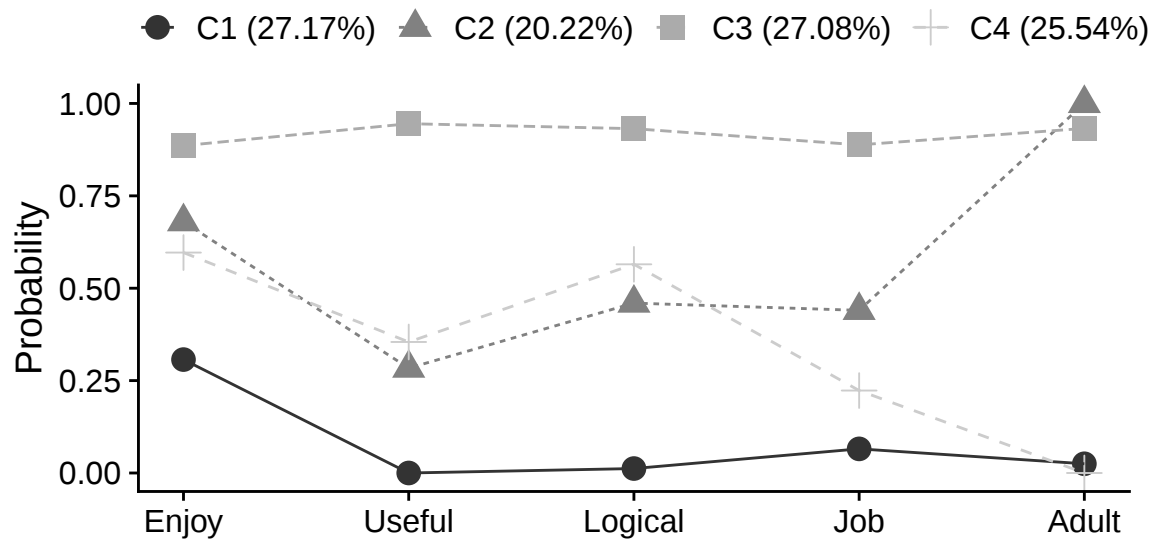
```

Time 1 LCA - Conditional Item Probability Plot

```

plot_lca_function(
  model_name = model_t1_c4,
  item_num = 5,
  class_num = 4,
  item_labels = c("Enjoy","Useful","Logical","Job","Adult"),
  plot_title = ""
)

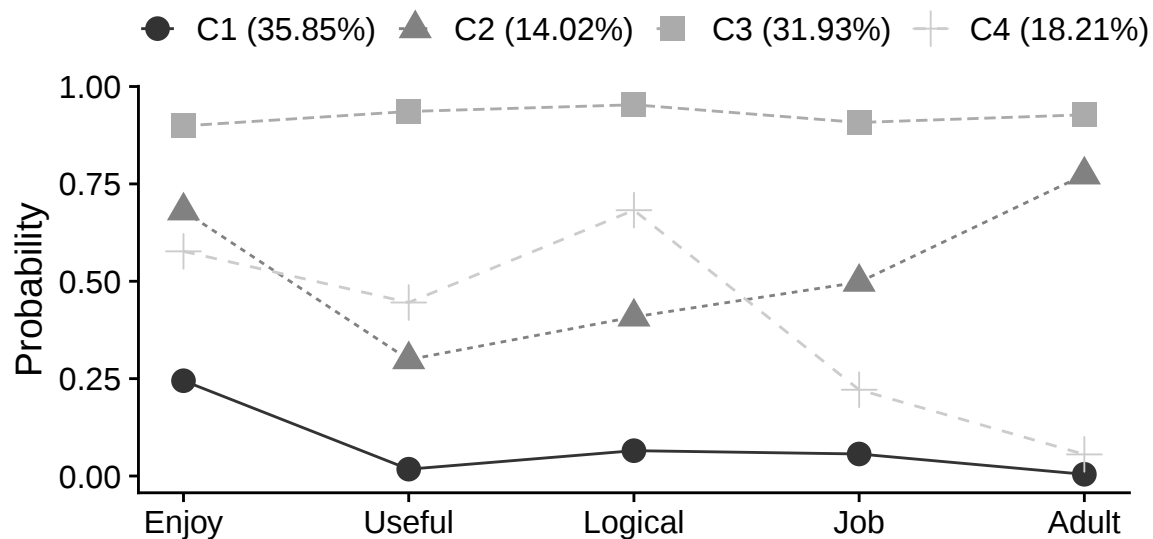
```



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

Time 2 LCA - Conditional Item Probability Plot

```
plot_lca_function(
  model_name = model_t2_c4,
  item_num = 5,
  class_num = 4,
  item_labels = c("Enjoy", "Useful", "Logical", "Job", "Adult"),
  plot_title = ""
)
```



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

Estimate Latent Transition Analysis Models (LTA)

Estimate Non-Invariant LTA Model

```
lta_non_inv <- mplusObject(

  TITLE =
    "LTA (Non-Invariant)",

  VARIABLE =
    "usevar = 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 allow item thresholds !!!
      !!! to be estimated for each class (e.g. noninvariance) !!!

    %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", "lta-non-invariant.inp"),
                               check=TRUE, run = TRUE, hashfilename = FALSE)

```

Estimate Invariant LTA Model

```

lta_inv <- mplusObject(

  TITLE =
    "LTA (Invariant)",

  VARIABLE =
    "usevar = 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:
    %c1#1%
    [AB39M$1-AB39X$1] (1-5); !!! labels that are repeated will constrain parameters to 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", "lta-invariant.inp"),
                           check=TRUE, run = TRUE, hashfilename = FALSE)

```

Conduct Sattorra-Bentler adjusted Log Likelihood Ratio Difference Testing

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

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

```
## <simpleError in file(file, "rt"): cannot open the connection>
```

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

```
# Log Likelihood Values
```

```
L0 <- lta_models[["lta.invariant.out"]][["summaries"]][["LL"]]
```

```
L1 <- lta_models[["lta.non.invariant.out"]][["summaries"]][["LL"]]
```

```
# LRT equation
```

```
lr <- -2*(L0-L1)
```

```
# Parameters
```

```
p0 <- lta_models[["lta.invariant.out"]][["summaries"]][["Parameters"]]
```

```
p1 <- lta_models[["lta.non.invariant.out"]][["summaries"]][["Parameters"]]
```

```
# Scaling Correction Factors
```

```
c0 <- lta_models[["lta.invariant.out"]][["summaries"]][["LLCorrectionFactor"]]
```

```
c1 <- lta_models[["lta.non.invariant.out"]][["summaries"]][["LLCorrectionFactor"]]
```

```

# Difference Test Scaling correction
cd <- ((p0*c0)-(p1*c1))/(p0-p1)

# Chi-square difference test(TRd)
TRd <- (lr)/(cd)

# Degrees of freedom
df <- abs(p0 - p1)

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

```

RESULT: The Log Likelihood χ^2 difference test comparing the invariant and non-invariant LTA models was, $\chi^2(20) = 18.23, p = .572$. [See Reference Here](#)

Alternative syntax: Run LTA using the createMixtures function (less verbose).

```

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")

```

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, or testing the equivalence of specific transition probabilities.

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

```

lta_invar <- lta_models$lta.invariant.out

par <- as_tibble(lta_invar[["parameters"]][["unstandardized"]]) %>%
  select(1:3) %>%

```

```
filter(grepl('ON|Means', paramHeader)) %>%
mutate(est = as.numeric(est))
```

```
# Name each parameter individually to make the subsequent calculations 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("Anti-Science", "Amb. w/ Elevated", "Amb. w/ Minimal", "Pro-Science"),
  "Anti-Science" = c(t11, t21, t31, t41),
  "Amb. w/ Elevated" = c(t12, t22, t32, t42),
  "Amb. w/ Minimal" = c(t13, t23, t33, t43),
  "Pro-Science" = 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) %>%
  tab_footnote(
    footnote = md("*Note.* Transition matrix values are identical to Table 5. However,
    Table 5 has the values rearranged by class for interpretation purposes. Classes may be
    arranged directly through Mplus syntax using start values."),
    locations = cells_title()
  )
```

Student transitions from 7th grade (rows) to 10th grade (columns)¹

7th grade	10th grade			
	Anti-Science	Amb. w/ Elevated	Amb. w/ Minimal	Pro-Science
Anti-Science	0.56	0.09	0.16	0.19
Amb. w/ Elevated	0.27	0.26	0.15	0.32
Amb. w/ Minimal	0.35	0.08	0.30	0.26
Pro-Science	0.21	0.14	0.12	0.52

¹Note. Transition matrix values are identical to Table 5. However, Table 5 has the values rearranged by class for interpretation purposes. Classes may be arranged directly through Mplus syntax using start values.

Estimate RI-LTA Model (Random Intercept As Continuous Factor)

Invariant RI-LTA Model

```
ri_lta <- mplusObject(

  TITLE =
    "RI-LTA (Invariant)",

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

    categorical =AB39M AB39T AB39U AB39W AB39X
      GA33A GA33H GA33I GA33K GA33L;

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

  ANALYSIS =
    "type = mixture;
    starts = 1000 250;
    process = 16(starts);
    algorithm = integration;",

  MODEL =
    "%overall%

    f by AB39M - AB39X* (p1-p5) !!! specify random intercept as continuous latent factor !!!
      GA33A - GA33L* (p1-p5);

    f@1; [f@0]; !!! fix random intercept variance to 1 and mean to 0 !!!

    c2 on c1;

    MODEL c1:
    %c1#1%
    [AB39M$1-AB39X$1] (1-5);

    %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);",

PLOT =
"type = plot3;
series= AB39M-AB39X (*);",

usevariables = colnames(lsay_data),
rdata = lsay_data)

ri_lta_fit <- mplusModeler(ri_lta,
  dataout=here("RI-LTA", "ri_lta.dat"),
  modelout=here("RI-LTA", "ri_lta_invariant.inp"),
  check=TRUE, run = TRUE, hashfilename = FALSE)

```

Compare Model FIT: Invariant LTA, Non-Invariant LTA, & Invariant RI-LTA Models

Read & extract model fit data for comparison

```

# Read models
lta_noninvar <- lta_models$lta.non.invariant.out
ri_lta <- readModels(here("RI-LTA", "ri-lta-invariant.out"), quiet = TRUE)

#Extract values for table
enum_extract1 <- LatexSummaryTable(lta_noninvar,
  keepCols=c("Title", "Parameters", "LL", "BIC", "aBIC", "Observations"))
enum_extract2 <- LatexSummaryTable(lta_invar,
  keepCols=c("Title", "Parameters", "LL", "BIC", "aBIC", "Observations"))
enum_extract3 <- LatexSummaryTable(ri_lta,
  keepCols=c("Title", "Parameters", "LL", "BIC", "aBIC", "Observations"))

```

Calculate indices derived from the Log Likelihood (LL)

```

allFit <- rbind(enum_extract1, enum_extract2, enum_extract3) %>%
  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) %>%
  select(1:5)

```

Format fit table

```
allFit %>%
  gt() %>%
  tab_header(
    title = md("**Model Fit Comparision Table**"), subtitle = md("&nbsp;")) %>%
  cols_label(
    Title = "Model",
    Parameters = md("Par"),
    LL = md("*LL*"),
    BIC = md("BIC"),
    aBIC = md("aBIC")) %>%
  tab_footnote(
    footnote = md(
      "*Note.* Par = Parameters;
      *LL* = model log likelihood;
      BIC = Bayesian information criterion;
      aBIC = sample size adjusted BIC."),
    locations = cells_title()) %>%
  tab_options(column_labels.font.weight = "bold")
```

Model Fit Comparision Table¹

Model	Par	<i>LL</i>	BIC	aBIC
LTA (Non-Invariant)	55	-14447.71	29336.88	29162.13
LTA (Invariant)	35	-14458.24	29197.40	29086.19
RI-LTA (Invariant)	40	-14442.02	29205.10	29078.00

¹Note. Par = Parameters; LL = model log likelihood; BIC = Bayesian information criterion; aBIC = sample size adjusted BIC.

Extract Invariant RI-LTA Mpdel Transitions

```
ri_lta_out <- ri_lta[["class_counts"]][["transitionProbs"]][["probability"]] %>%
  as.data.frame(as.numeric())

ri_t_matrix <- tibble(
  "Time1" = c("Anti-Science", "Amb. w/ Elevated", "Pro-Science", "Amb. w/ Minimal"),
  "Anti-Science" = c(ri_lta_out[1,1], ri_lta_out[2,1], ri_lta_out[3,1], ri_lta_out[4,1]),
  "Amb. w/ Elevated" = c(ri_lta_out[5,1], ri_lta_out[6,1], ri_lta_out[7,1], ri_lta_out[8,1]),
  "Pro-Science" = c(ri_lta_out[9,1], ri_lta_out[10,1], ri_lta_out[11,1], ri_lta_out[12,1]),
  "Amb. w/ Minimal" = c(ri_lta_out[13,1], ri_lta_out[14,1], ri_lta_out[15,1], ri_lta_out[16,1]))
```

Create table

```
ri_t_matrix %>%
  gt(rowname_col = "Time1") %>%
```

```

tab_stubhead(label = "7th grade") %>%
tab_header(
  title = md("**Random Intercept Model Transition Matrix**"),
  subtitle = md("**Student transitions from 7th grade (rows) to 10th grade (columns)**")) %>%
fmt_number(2:5, decimals = 2) %>%
tab_spanner(label = "10th grade", columns = 2:5) %>%
  tab_footnote(
    footnote = md(
      "*Note.* Transition matrix values are the identical to Table 5. However Table 5
      values are rearranged by class for interpretation purposes. Classes may be arranged
      directly through Mplus syntax using start values."),
    locations = cells_title()
  )

```

Random Intercept Model Transition Matrix¹
 Student transitions from 7th grade (rows) to 10th grade (columns)

7th grade	10th grade			
	Anti-Science	Amb. w/ Elevated	Pro-Science	Amb. w/ Minimal
Anti-Science	0.54	0.10	0.18	0.18
Amb. w/ Elevated	0.24	0.25	0.33	0.18
Pro-Science	0.21	0.14	0.51	0.14
Amb. w/ Minimal	0.35	0.09	0.25	0.31

¹Note. Transition matrix values are the identical to Table 5. However Table 5 values are rearranged by class for interpretation purposes. Classes may be arranged directly through Mplus syntax using start values.

Invariant & Non-Invariant LTA Conditional Item Probability Plots

Create a function for plotting the conditional item probabilities estimated from an LTA model.

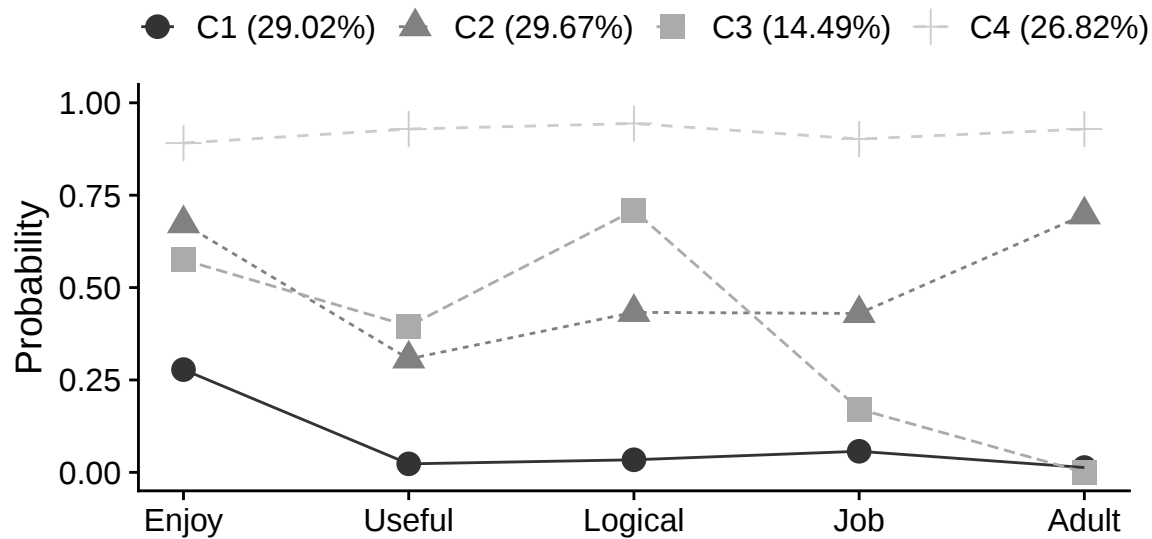
The `plot_lta_function` requires one additional argument called `timepoint` used to specify the time point to extract probabilities (e.g., 1).

```
plot_lta_function <- function(model_name,item_num,class_num,timepoint,item_labels,plot_title){  
  
  # Extract Item Probabilities  
  mplus_model <- as_tibble(model_name$parameters$probability.scale) %>%  
    filter(category=="2", str_detect(LatentClass, glue("C{timepoint}"))) %>%  
    select(LatentClass,est, param) %>%  
    pivot_wider(names_from = LatentClass, values_from = est) %>%  
    select(-param)  
  
  # Create class size in percentages (%)  
  c_size <- as.data.frame(model_name$class_counts$modelEstimated) %>%  
    filter(str_detect(variable, glue("C{timepoint}"))) %>%  
    select(proportion)  
  colnames(c_size) <- paste0("cs")  
  c_size <- c_size %>% mutate(cs = round(cs * 100, 2))  
  colnames(mplus_model) <- paste0("C", 1:class_num, glue(" ({c_size[1:class_num,]}%)"))  
  
  # Variable names  
  plot_t1 <- cbind(Var = paste0("U", 1:item_num), mplus_model)  
  plot_t1$Var <- factor(plot_t1$Var,  
    labels = item_labels)  
  plot_t1$Var <- fct_inorder(plot_t1$Var)  
  pd_long_t1 <- melt(plot_t1, id.vars = "Var")  
  
  p <- 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:item_num, labels = plot_t1$Var) +  
    scale_colour_grey() + scale_y_continuous(limits = c(0,1)) +  
    labs(title = plot_title, y = "Probability") +  
    theme_cowplot() +  
    theme(legend.title = element_blank(),  
      legend.position = "top")  
  
  p  
  return(p)  
}
```

Invariant LTA Model - Conditional Item Probability Plot

For the invariant LTA model, conditional item probabilities are the same across time-points.

```
plot_lta_function(  
  model_name = lta_invar,  
  item_num = 5,  
  class_num = 4,  
  timepoint = 1,  
  item_labels = c("Enjoy", "Useful", "Logical", "Job", "Adult"),  
  plot_title = ""  
)
```



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

Non-Invariant LTA - Conditional Item Probability Plot (Timepoint 1)

```
T1 <- plot_lta_function(  
  model_name = lta_noninvar,  
  item_num = 5,  
  class_num = 4,  
  timepoint = 1,  
  item_labels = c("Enjoy", "Useful", "Logical", "Job", "Adult"),  
  plot_title = "Timepoint 1")
```

Non-Invariant RI-LTA - Conditional Item Probability Plot (Timepoint 2)

```
T2 <- plot_lta_function(  
  model_name = lta_noninvar,  
  item_num = 5,  
  class_num = 4,
```

```
timepoint = 2,
item_labels = c("Enjoy", "Useful", "Logical", "Job", "Adult"),
plot_title = "Timepoint 2")
```

```
# combine plots with the `patchwork` package
```

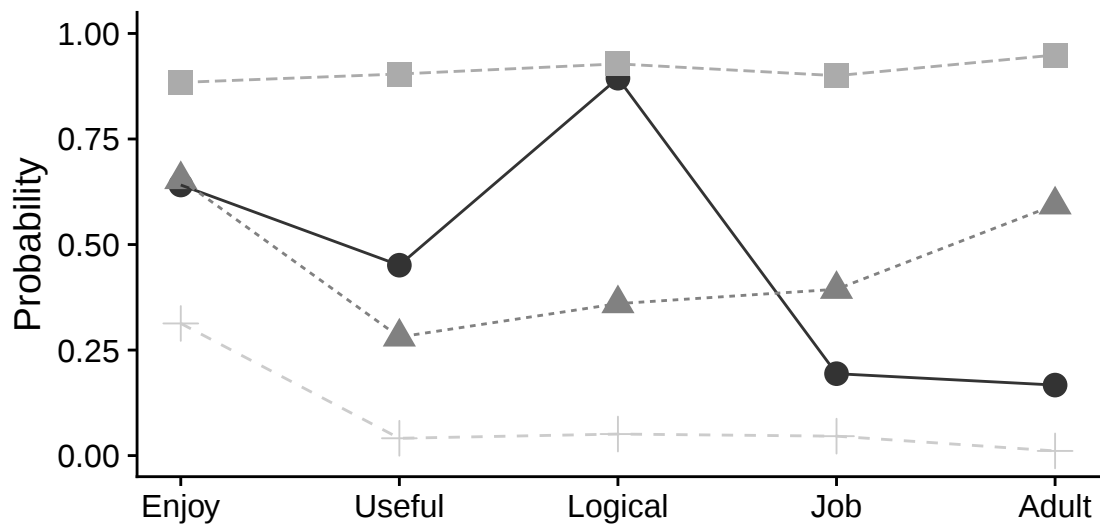
```
T1 / T2 +
```

```
plot_annotation(title = 'Non-Invariant LTA Conditional Item Probability Plots')
```

Non-Invariant LTA Conditional Item Probability Plots

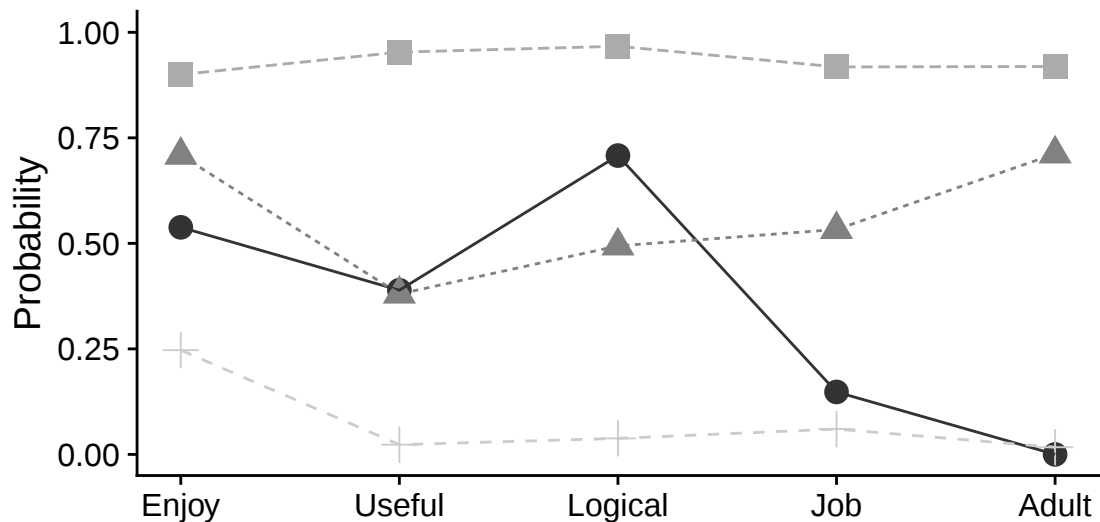
Timepoint 1

● C1 (12.74%) ▲ C2 (28.83%) ■ C3 (27.98%) + C4 (30.46%)



Timepoint 2

● C1 (15.83%) ▲ C2 (18.68%) ■ C3 (30.21%) + C4 (35.28%)



```
ggsave(here("figures", "Non_Inv_LTA_plots_T1-T2.png"), dpi=300, height=10, width=12, units="in")
```

LTA Transition Plot

This code is adapted from the source code for the `plotLTA` function found in the [MplusAutomation](#) package

NOTE: The function found in `plot_transitions_function.R` is specific to a model with 2 time-points and 4-classes & must be updated to accommodate other models.

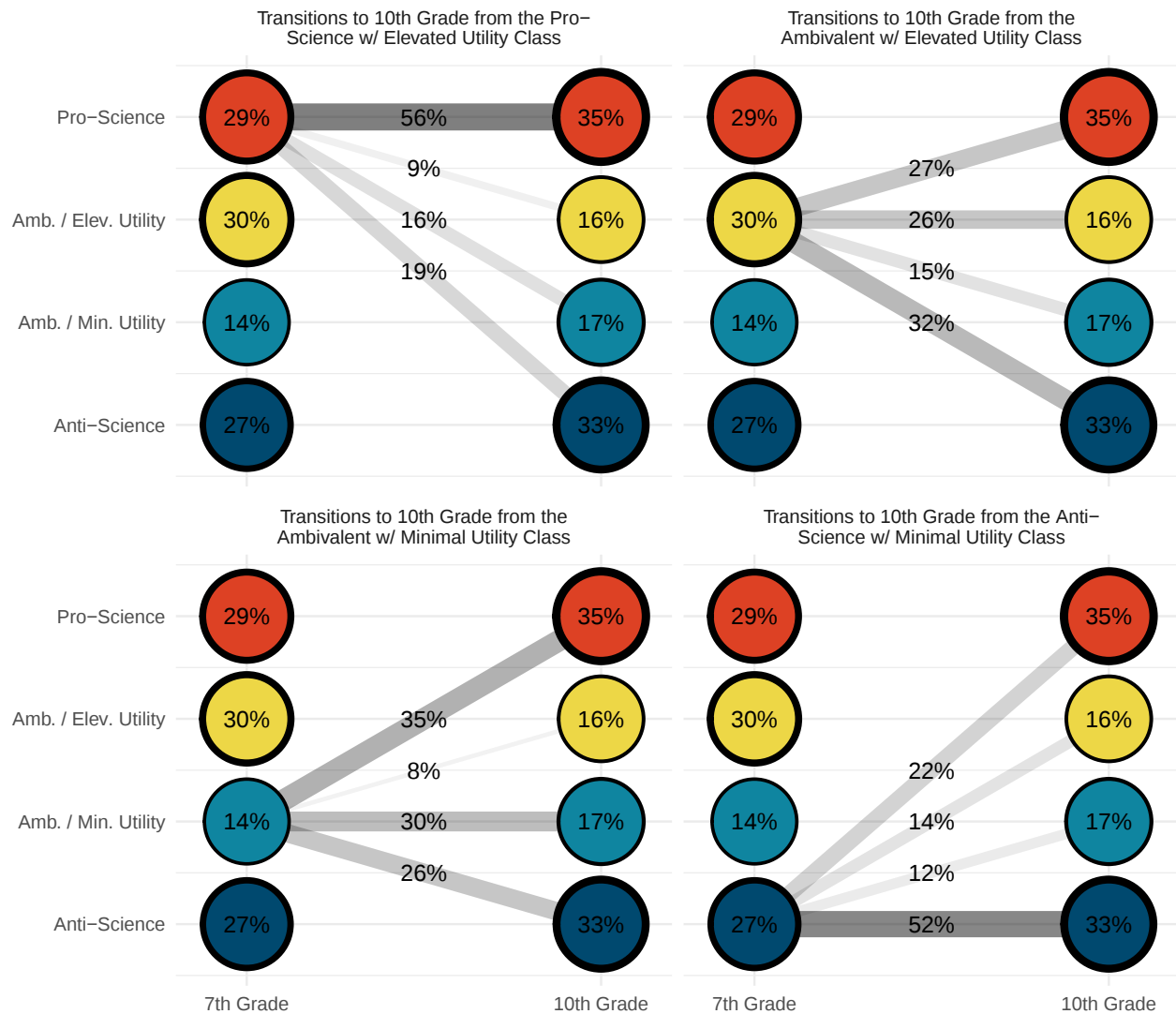
```
library(PNWColors)
library(ggrepel)

source("plot_transitions_function.R") # Script is located in the project repository

lta_model <- readModels(here("LTA_models", "lta-invariant.out"), quiet = TRUE)

## <simpleError in file(file, "rt"): cannot open the connection>

plot_transitions_function(
  model_name = lta_model,
  color_pallette = pnw_palette("Bay", n=4, type = "discrete"),
  facet_labels = c(
    `1` = "Transitions to 10th Grade from the Pro-Science w/ Elevated Utility Class",
    `2` = "Transitions to 10th Grade from the Ambivalent w/ Elevated Utility Class",
    `3` = "Transitions to 10th Grade from the Ambivalent w/ Minimal Utility Class",
    `4` = "Transitions to 10th Grade from the Anti-Science w/ Minimal Utility Class"),
  timepoint_labels = c('1' = "7th Grade", '2' = "10th Grade"),
  class_labels = c(
    "Pro-Science",
    "Amb. / Elev. Utility",
    "Amb. / Min. Utility",
    "Anti-Science")
)
```



```
ggsave(here("figures", "LTA_transition_plot.png"), dpi=500, height=7, width=8, units="in")
```

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, B., & Asparouhov, T. (2020). Latent transition analysis with random intercepts (RI-LTA). *Psychological Methods*. Advance online publication. <https://doi.org/10.1037/met0000370>

Muthén L.K., & Muthen B.O. (1998-2017) Mplus User's Guide. Eight Edition. Los Angeles, CA: Muthen & Muthen.

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 H, Averick M, Bryan J, Chang W, McGowan LD, François R, Golemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). “Welcome to the tidyverse.” *Journal of Open Source Software*, 4(43), 1686. doi: 10.21105/joss.01686.