Appendix - LTA with MplusAutomation

Contents

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` (cite) using the wrapping program `MplusAutomation` (cite). This method requires that the user to have the proprietary software `Mplus` installed on their OS.

This approach 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.

Preparation

Download the R-Project

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

On the Github repository webpage:

a. Click the green `Code` menu button and choose `Download ZIP`
b. For Github users, `fork` your own `branch` of the lab repository to create a version controlled project

Project folder organization: nested structure

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

1. "data"
2. "enum_LCA_time1"
3. "enum_LCA_time2"
4. "LTA_models"

Note regarding choosing the project location:

If the 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`.

- **Type 1 comment:** The hashtag symbol `#` identifies comments written in R-language form.
- **Type 2 comment:** Within the `mplusObject()` function all text used to generate Mplus input files is enclosed within quotation marks (green text). To add comments use the Mplus language comment convention e.g., (!!! annotate Mplus input !!!).
- **Type 3 comment:** To signal to the user areas of the syntax which must be adapted to fit specific modeling contexts the text, `NOTE CHANGE:` is used.

Load packages

```
library(MplusAutomation) # package descriptions to be added
library(tidyverse)       # package descriptions to be added
library(here)            # package descriptions to be added
library(glue)            # package descriptions to be added
library(janitor)         # package descriptions to be added
library(gt)              # package descriptions to be added
```

Read in LSAY data file (CSV format)

Note: LSAY data 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: '6' indicates the number of k-class models to estimate.
# User can change this number it 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 chunk.
    TITLE = glue("Class-{k}_Time1"),

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

     !!! The value of 'k' is inserted here !!!
     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_tim
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)
})
```

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

     classes = c({k}); !!! Loop value 'k' inserted here !!!"),

  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)
})
```

Step 2: Create model fit summary table

Read models and extract data to compose the model fit table

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

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(" ")) %>%
    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 
    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[1]

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

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

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

Step 3: Estimate Latent Transition Analysis

Run freely estimated LTA model (non-invariant)

```
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;",

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

```
      ! This is the same syntax as written below:
      ! 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:
      %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("enum_LCA_time2", "lta.dat"),
                    modelout=here("LTA_models", "4-Class-Non-Invariant.inp"),
                    check=TRUE, run = TRUE, hashfilename = FALSE)
```

Run 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;",

  MODEL =
     "%overall%
      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);",

  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("enum_LCA_time2", "lta.dat"),
                 modelout=here("LTA_models", "4-Class-Invariant.inp"),
                 check=TRUE, run = TRUE, hashfilename = FALSE)
```

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

```
# 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_header(
    title = md("**Student transitions from 7th grade (rows) to 10th grade (columns)**"), subtitle = md(
  fmt_number(2:5,decimals = 2) %>%
  tab_spanner(label = "10th grade",columns = 2:5)
```

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

	10th grade			
	C2=1	C2=2	C2=3	C2=4
C1=1	0.52	0.21	0.12	0.15
C1=2	0.19	0.56	0.16	0.09
C1=3	0.26	0.35	0.30	0.08
C1=4	0.32	0.27	0.15	0.27

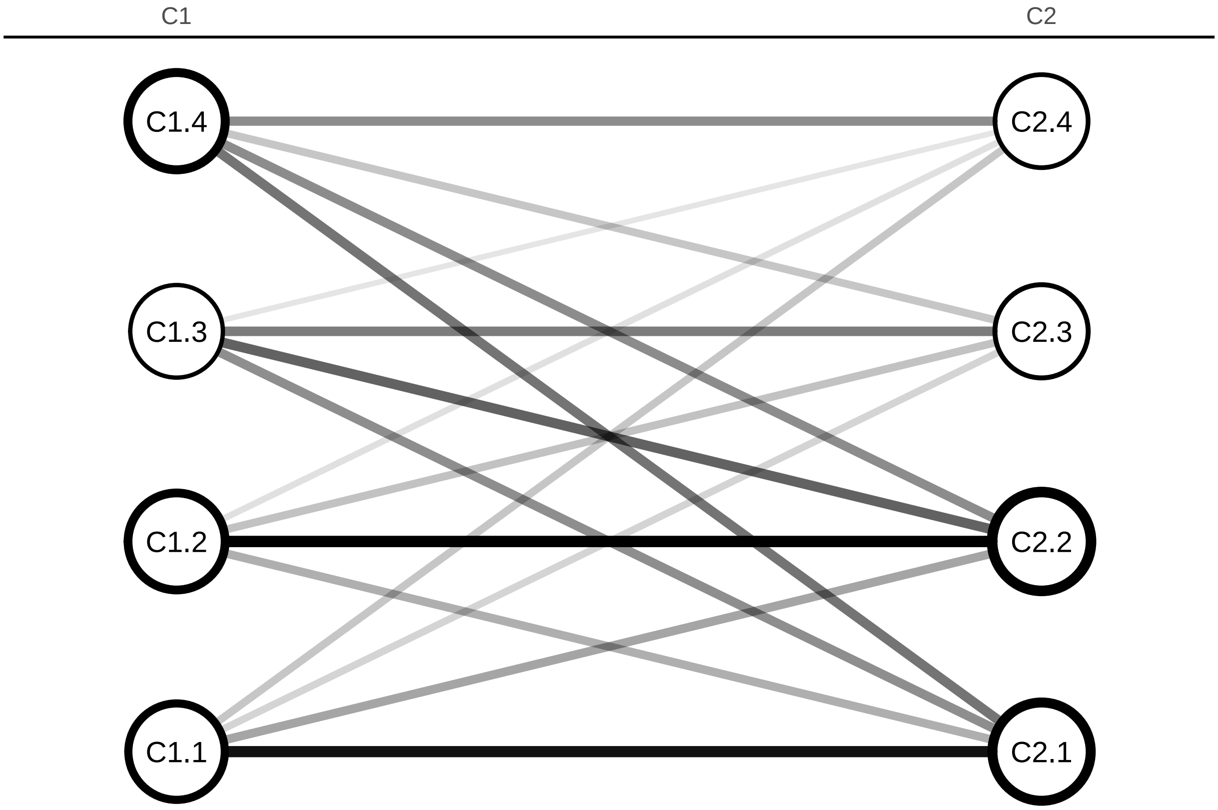
END COPY EDITING - ALL PLOTS CURRENTLY INCOMPLETE

Plot LTA transitions

Type 1 LTA plot: use `plotLTA`

- issue1: node proportions & transition proportions missing
- issue2: change to faceted plot

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



Type 2 LTA plot: sankey interactive chart (INCOMPLETE)

- issue 1: Proportions at time 1 are incorrect. Unclear how to incorporate initial class size data into chart.
- issue 2: Unclear how to label node and transition. Examples to work from are unavailable.

Change to long-format

```
trans_long <- t_matrix %>%
  pivot_longer(`C2=1`:`C2=4`, # The columns I'm gathering together
           names_to = "Time2", # new column name for existing names
         values_to = "value") # new column name to store values

nodes <- data.frame(name=c(as.character(trans_long$Time1),
                           as.character(trans_long$Time2)) %>%
                      unique())

trans_long$IDTime1=match(trans_long$Time1, nodes$name)-1
trans_long$IDTime2=match(trans_long$Time2, nodes$name)-1
```

Prepare colour scale

```
ColourScal ='d3.scaleOrdinal().range([
"#FDE725FF","#B4DE2CFF","#6DCD59FF","#35B779FF","#1F9E89FF",
"#26828EFF","#31688EFF","#3E4A89FF","#482878FF","#440154FF"])'
```

```r
#library(networkD3)

sankeyNetwork(Links = trans_long, Nodes = nodes,
              Source = "IDTime1", Target = "IDTime2",
              Value = "value", NodeID = "name",
              sinksRight=FALSE, colourScale=ColourScal,
              nodeWidth=40, fontSize=13, nodePadding=20)
```

Type 3 LTA plot: `alluvial` (INCOMPLETE)

- issue 1: Proportions at time 1 are incorrect. Unclear how to incorporate initial class size data into chart.
- issue 2: Unclear how to label node and transitions in a clear manner. Examples to work from are unavailable.

```r
library(ggalluvial)
#library(gghighlight)

ggplot(trans_long,
       aes(axis1 = Time1, axis2 = Time2,
           y = value)) +
  scale_x_discrete(limits = c("7th Grade", "10th Grade"), expand = c(.2, .05)) +
  geom_alluvium(aes(fill = Time1), show.legend = FALSE) +
  geom_stratum() +
  #geom_text(stat = "alluvium", aes(label = round(value,2), color = Time1),
  #          show.legend = FALSE, fontface = 2, position = position_nudge(x = -0.21)) +
  theme_minimal()
  # + gghighlight(label_key = value)
```

```r
library(parcats)
library(easyalluvial)

p = alluvial_wide(t_matrix, max_variables = 5)

parcats(p, marginal_histograms = TRUE, data_input = t_matrix)
```

References: (INCOMPLETE)

Hallquist, Michael N., and Joshua F. Wiley. 2018. "MplusAutomation: An R Package for FacilitatingLarge-Scale Latent Variable Analyses in Mplus."Structural Equation Modeling, 1–18. https://doi.org/10.1080/10705511.2017.1402334.

Müller, Kirill. 2017.Here: A Simpler Way to Find Your Files. https://CRAN.R-project.org/package=here.

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

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

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

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