Appendix A

Rationale for MplusAutomation workflow:

This R code tutorial is intended to provide a template for running LCA, auxiliary variable integration, and figure generation 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 the user to have the proprietary software `Mplus` installed on their computer.

This approach also relies on the utility of `R-Projects`. This provides a structured framework for organizing all associated data files, Mplus input/output files, scripts, and figures. Given the high output of Mplus files inherent to mixture 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

Link to Github repository here: https://github.com/garberadamc/LCA-COPING-BD

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 named `LCA-COPING-BD.Rproj`
4. Next open the file containing all analysis code named `rcode-lca-coping-bd.R`.

Note: 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_mplus"; 3. "3step_mplus"; 4. "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`.

To install package {`rhdf5`}

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

BiocManager::install("rhdf5")
```

Load packages

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

File-paths are set using R-Projects & the {`here`} package

Read in data

```
ies_data <- read_csv(here("data","ies_bd_data.csv"),
                     na=c("",".", "999"))
```

Enumeration: compare *k*-class models 1 - 6

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

    TITLE = glue("Class-{k} LCA Enumeration - Youth Coping Strategies"),

    VARIABLE =
  glue(
    "categorical = do1 do2 do3 do5 do6;
     usevar = do1 do2 do3 do5 do6;

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

```
  ANALYSIS =
   "estimator = mlr;
    type = mixture;
    processors=10;",

  PLOT =
    "type = plot3;
    series = do1 do2 do3 do5 do6(*);",

  OUTPUT = "sampstat tech11 tech14;",

  usevariables = colnames(ies_data),
  rdata = ies_data)

lca_enum_fit <- mplusModeler(lca_enum,
                  dataout=glue(here("enum_mplus", "c_lca_enum_bd.dat")),
                  modelout=glue(here("enum_mplus", "c{k}_lca_enum_bd.inp")) ,
                  check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

Compare model fit for enumerated models

```
output_enum <- readModels(here("enum_mplus"), quiet = TRUE)

enum_extract <- LatexSummaryTable(output_enum,
                keepCols=c("Title", "Parameters", "LL", "BIC", "aBIC",
                           "BLRT_PValue", "T11_VLMR_PValue","Observations"))

allFit <- enum_extract %>%
  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 %>%
  mutate(Title = str_remove(Title, " LCA Enumeration - Youth Coping Strategies")) %>%
  gt() %>%
  tab_header(
    title = md("**Model Fit Summary Table**"), subtitle = md(" ")) %>%
  cols_label(
    Title = "Classes",
    Parameters = md("Par"),
    LL = md("*LL*"),
    T11_VLMR_PValue = "VLMR",
    BLRT_PValue = "BLRT",
    BF = md("BF"),
    cmPk = md("*cmP_k*")) %>%
  tab_footnote(
```

```
    footnote = md(
    "*Note.* Par = parameters; *LL* = log likelihood;
      BIC = bayesian information criterion;
      aBIC = sample size adjusted BIC; CAIC = consistent Akaike information criterion;
      AWE = approximate weight of evidence criterion;
      BLRT = bootstrapped likelihood ratio test p-value;
      VLMR = Vuong-Lo-Mendell-Rubin adjusted likelihood ratio test p-value;
      cmPk = approximate correct model probability."),
    locations = cells_title()) %>%
  tab_options(column_labels.font.weight = "bold") %>%
  fmt_number(10,decimals = 2,
             drop_trailing_zeros=TRUE,
             suffixing = TRUE) %>%
  fmt_number(c(3:9,11),
             decimals = 0) %>%
  fmt_missing(1:11,
              missing_text = "--") %>%
  fmt(c(8:9,11),
    fns = function(x)
    ifelse(x<0.001, "<.001",
           scales::number(x, accuracy = 0.01))) %>%
  fmt(10, fns = function(x)
    ifelse(x>100, ">100",
           scales::number(x, accuracy = .1)))
```

Model Fit Summary Table[1]

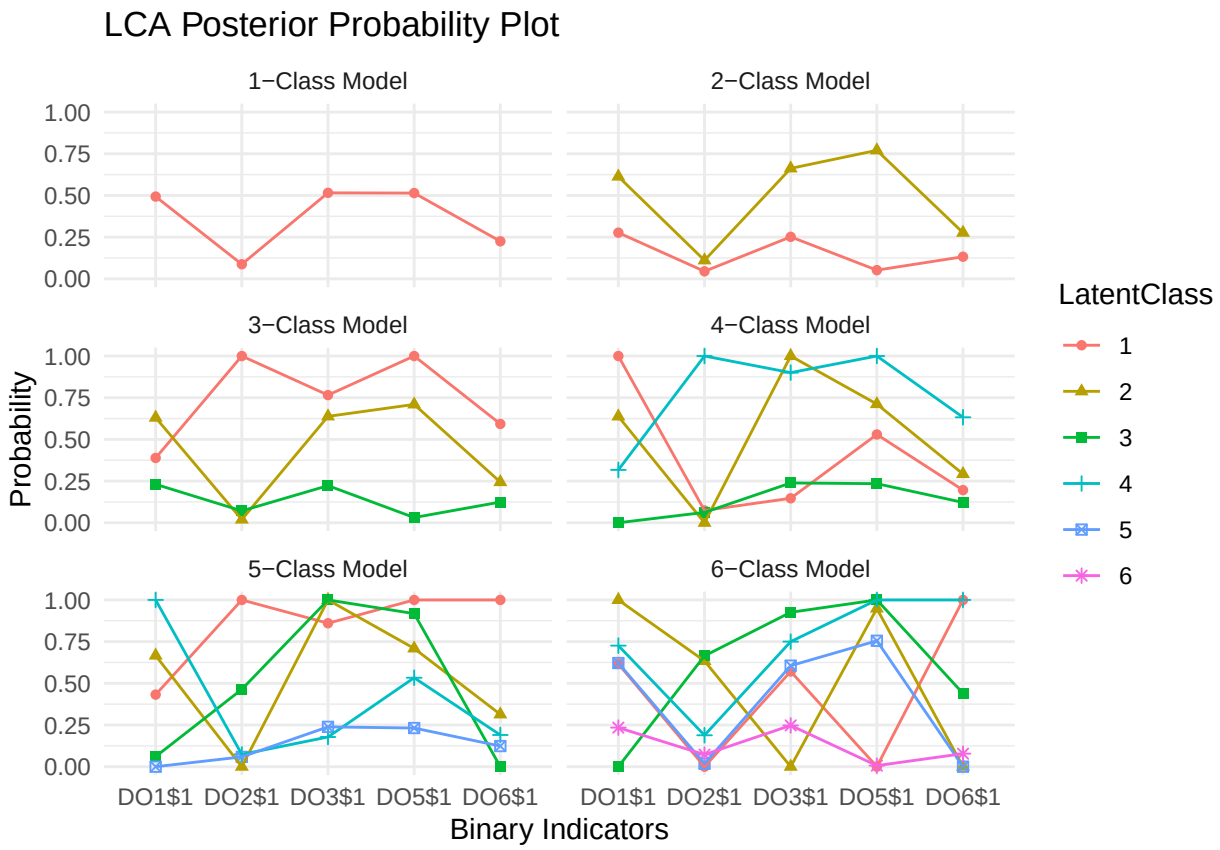
Classes	Par	*LL*	BIC	aBIC	CIAC	AWE	BLRT	VLMR	BF	*cmP_k*
Class-1	5	−1,294	2,619	2,603	2,624	2,664	–	–	>100	<.001
Class-2	11	−1,259	2,586	2,551	2,597	2,686	<.001	<.001	>100	1.00
Class-3	17	−1,249	2,602	2,548	2,619	2,757	<.001	0.12	>100	<.001
Class-4	23	−1,244	2,629	2,556	2,652	2,838	0.33	0.01	>100	<.001
Class-5	29	−1,242	2,662	2,570	2,691	2,926	0.67	0.34	>100	<.001
Class-6	35	−1,241	2,695	2,584	2,730	3,014	0.67	0.50	–	<.001

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

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

```
model_results <- data.frame()

for (i in 1:length(output_enum)) {

  temp <- output_enum[[i]]$parameters$unstandardized

  temp <- data.frame(unclass(temp))%>%
```

```
    mutate(model = paste0(i, "-Class Model"))

  model_results <- rbind(model_results, temp)
}

rm(temp)

model_results <- model_results %>%
  filter(paramHeader == "Thresholds") %>%
  select(est, model, LatentClass, param) %>%
  mutate(prob = (1 / (1 + exp(est)))) %>%
  select(-est)

ggplot(model_results,
       aes(x = param, y = prob,
           color = LatentClass,
           shape = LatentClass,
           group = LatentClass)) +
  geom_point() + geom_line() +
  facet_wrap(~ model, ncol = 2) +
  labs(title = "LCA Posterior Probability Plot",
       x= "Binary Indicators", y = "Probability") +
  theme_minimal()
```



```
# save plot
ggsave(here("figures","C1-6_LCA_PP-plot.png"),
       dpi=300, height=4, width=6, units="in")
```

Manual "3-Step" ML Auxiliary Variable Integration Method

Step 1 - Estimate the unconditional model & list all auxiliary variables

```
m_step1  <- mplusObject(
  TITLE = "Step1_3step_automation Behavioral Disorder",
  VARIABLE =
   "categorical = do1 do2 do3 do5 do6;

    usevar = do1 do2 do3 do5 do6;

    classes = c(3);

    !!! All potential auxiliary variables should be listed here !!!
    auxiliary =
    FEMALE ETHN_CMP SOC_STRS
    BOTHR_U negmood1 posmood1;",

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

  SAVEDATA =
   "!!! This saved dataset will contain class probabilities and modal assignment columns !!!
    File=3step_BD_savedata_012020.dat;
    Save=cprob;
    Missflag= 999;",

  MODEL = "",
  OUTPUT = "",

  PLOT =
    "type = plot3;
    series = do1 do2 do3 do5 do6(*);",

  usevariables = colnames(ies_data),
  rdata = ies_data)

m_step1_fit <- mplusModeler(m_step1,
                 dataout=here("3step_mplus", "Step1_3step_BD.dat"),
```

```
                  modelout=here("3step_mplus", "Step1_3step_BD.inp") ,
                  check=TRUE, run = TRUE, hashfilename = FALSE)
```

Step 2 - extract logits & saved data from the step 1 model

Extract logits for the classification probabilities for the most likely latent class

```
logit_cprobs <- as.data.frame(m_step1_fit[["results"]]
                                          [["class_counts"]]
                                          [["logitProbs.mostLikely"]])
```

Extract saved data from the step 1 model `mplusObject` named "m_step1_fit"

```
savedata <- as.data.frame(m_step1_fit[["results"]]
                                      [["savedata"]])
```

Rename the column in savedata for "C" and change to "N"

```
colnames(savedata)[colnames(savedata)=="C"] <- "N"
```

Estimate step 2 model

```
m_step2  <- mplusObject(
  TITLE = "Step2_3step_automation Behavioral Disorder",

  VARIABLE =
 "nominal=N;
  USEVAR = n;
  missing are all (999);
  classes = c(3); ",

  ANALYSIS =
 "estimator = mlr;
  type = mixture;
  starts = 0;",

  MODEL =
    glue(
 "%C#1%
  [n#1@{logit_cprobs[1,1]}];
  [n#2@{logit_cprobs[1,2]}];

  %C#2%
  [n#1@{logit_cprobs[2,1]}];
  [n#2@{logit_cprobs[2,2]}];
```

```
  %C#3%
  [n#1@{logit_cprobs[3,1]}];
  [n#2@{logit_cprobs[3,2]}];"),

  OUTPUT = "!tech11  tech14 res;",

  PLOT =
 "!type = plot3;
  !series = do1 do2 do3 do5 do6(*);",

  usevariables = colnames(savedata),
  rdata = savedata)

m_step2_fit <- mplusModeler(m_step2,
                 dataout=here("3step_mplus", "Step2_3step_BD.dat"),
                 modelout=here("3step_mplus", "Step2_3step_BD.inp"),
                 check=TRUE, run = TRUE, hashfilename = FALSE)
```

Estimate step 3 model

```
m_step3  <- mplusObject(
  TITLE = "Step3_3step_automation Behavioral Disorder",

  VARIABLE =
 "nominal = N;
  usevar = n;
  missing are all (999);

  usevar = SOC_STRS POSMOOD1 NEGMOOD1;
  classes = c(3); ",

  DEFINE =
 "Center SOC_STRS (Grandmean);
 !!! covariate is centered so that reported distal means are estimated at the weighted average of Social

  ANALYSIS =
 "estimator = mlr;
  type = mixture;
  starts = 0;",

  MODEL =
  glue(
 "!DISTAL = POSMOOD1 NEGMOOD1
  !MODERATOR = SOC_STRS

  %OVERALL%
  POSMOOD1 on SOC_STRS;
  POSMOOD1;
```

```
 NEGMOOD1 on SOC_STRS;
 NEGMOOD1;

 %C#1%
 [n#1@{logit_cprobs[1,1]}];
 [n#2@{logit_cprobs[1,2]}];

 [NEGMOOD1](m01);
 NEGMOOD1;
 NEGMOOD1 on SOC_STRS (s01);

 [POSMOOD1] (m1);
 POSMOOD1;
 POSMOOD1 on SOC_STRS (s1);

 %C#2%
 [n#1@{logit_cprobs[2,1]}];
 [n#2@{logit_cprobs[2,2]}];

 [NEGMOOD1](m02);
 NEGMOOD1;
 NEGMOOD1 on SOC_STRS (s02);

 [POSMOOD1] (m2);
 POSMOOD1;
 POSMOOD1 on SOC_STRS (s2);

 %C#3%
 [n#1@{logit_cprobs[3,1]}];
 [n#2@{logit_cprobs[3,2]}];

 [NEGMOOD1](m03);
 NEGMOOD1;
 NEGMOOD1 on SOC_STRS (s03);

 [POSMOOD1] (m3);
 POSMOOD1;
 POSMOOD1 on SOC_STRS (s3);"),

 MODELCONSTRAINT =
"New (diff12 diff13
 diff23 slope12 slope13
 slope23 ndiff12 ndiff13
 ndiff23 nslope12 nslope13
 nslope23);

 diff12 = m1-m2;
 diff13 = m1-m3;
 diff23 = m2-m3;
 slope12 = s1-s2;
 slope13 = s1-s3;
 slope23 = s2-s3;
```

```
  ndiff12 = m01-m02;
  ndiff13 = m01-m03;
  ndiff23 = m02-m03;
  nslope12 = s01-s02;
  nslope13 = s01-s03;
  nslope23 = s02-s03;",

  MODELTEST =
  ### NOTE: Only a single Wald test can be conducted at a time. ###
 "m1=m2;       !!! Distal outcome omnibus Wald test for `POSMOOD1` !!!
  m2=m3;

  !s1=s2;      !!! Slope difference omnibus Wald test `POSMOOD1 on SOC_STRS` !!!
  !s2=s3;

  !m01=m02;    !!! Distal outcome omnibus Wald test for `NEGMOOD1` !!!
  !m02=m03;

  !s01=s02;   !!! Slope difference omnibus Wald test for `POSMOOD1 on SOC_STRS` !!!
  !s02=s03;",

  usevariables = colnames(savedata),
  rdata = savedata)

m_step3_fit <- mplusModeler(m_step3,
                 dataout=here("3step_mplus", "Step3_3step_BD.dat"),
                 modelout=here("3step_mplus", "Step3_3step_BD.inp"),
                 check=TRUE, run = TRUE, hashfilename = FALSE)
```

End of 3-step procedure

Estimate step 3 model with covariate un-centered for simple-slopes plots

Note: Here the `update()` function is used to take the previous model and remove the Mplus syntax within the `DEFINE` statement that was used to center the covariate `Social Stress`. Next, the updated model input syntax is used to estimate a new model. To learn more about the `update` function see the `MplusAutomation` tutorial article (https://www.tandfonline.com/doi/pdf/10.1080/10705511.2017.1402334).

```
m_uncen <- update(m_step3,

  DEFINE = ~" ") # This update removes the centering syntax from the model object `m_step3`

m_uncen_fit <- mplusModeler(m_uncen,
                            dataout=here("3step_mplus", "Step3_3step_BD.dat"),
                            modelout=here("3step_mplus", "Step3_uncentered_BD.inp"),
                            check=TRUE, run = TRUE, hashfilename = FALSE)
```

Generate plots:

1. LCA Posterior probability plot
2. Distal outcome plots
3. Simple slope plots

LCA Posterior probability plot

Read in plot data from mplus output file `Step1_3step_BD.out`

```
model_step1 <- readModels(here("3step_mplus", "Step1_3step_BD.out"), quiet = TRUE)
```

This syntax creates a function called `plot_lca_function` that requires 5 arguments (inputs):

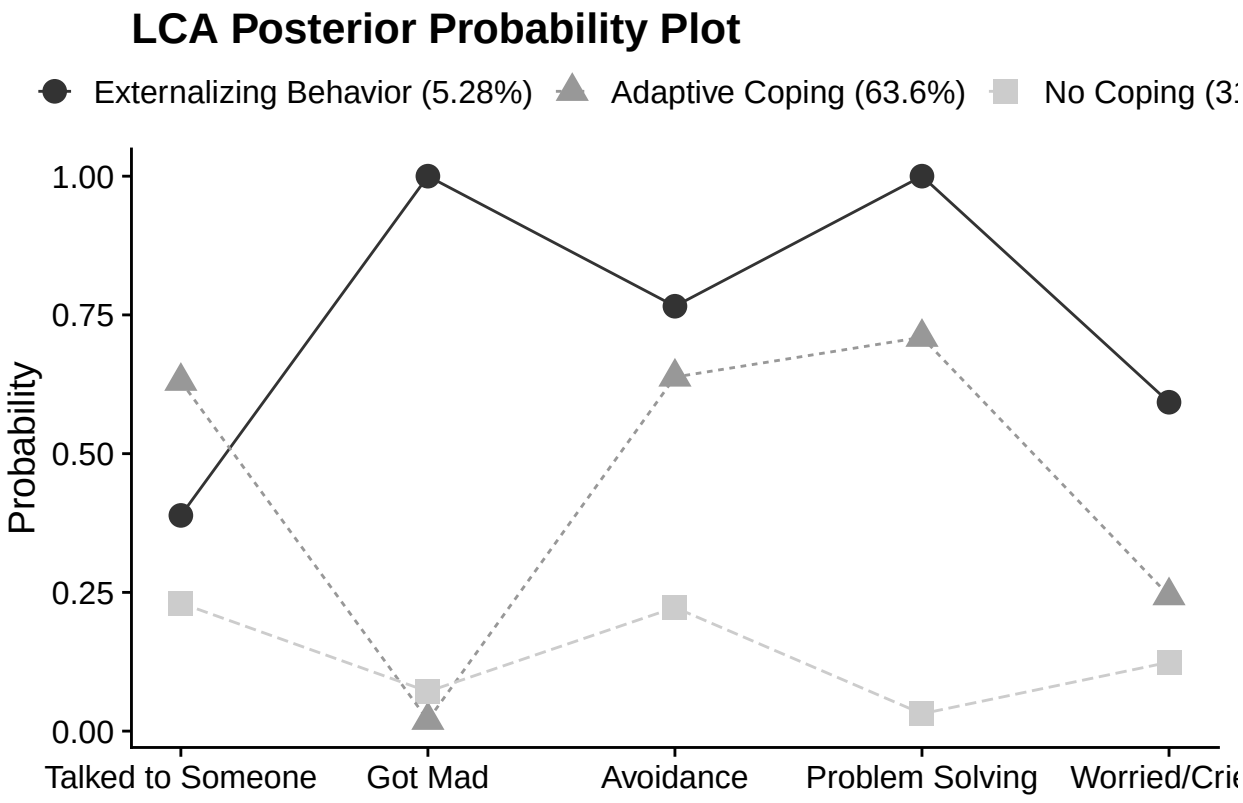
- `model_name`: name of Mplus model object (e.g., `model_step1`)
- `item_num`: the number of items in LCA measurement model (e.g., `5`)
- `class_num`: the number of classes (k) in LCA model (e.g., `3`)
- `item_labels`: the item labels for x-axis (e.g., `c("Enjoy","Useful","Logical","Job","Adult")`)
-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

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

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

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

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

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

```
p <- pd_long_data %>%
  ggplot(aes(x = as.integer(Var), y = value,
  shape = variable, colour = variable, lty = variable)) +
  geom_point(size = 4) + geom_line() +
  scale_x_continuous("", breaks = 1:5, labels = plot_data$Var) +
  scale_colour_grey() +
  labs(title = plot_title, y = "Probability") +
  theme_cowplot() +
  theme(legend.title = element_blank(),
        legend.position = "top")

p
return(p)
}
```

Run the `plot_lca_function` by specifying each input

```
plot_lca_function(
  model_name = model_step1,
  item_num = 5,
  class_num = 3,
  item_labels = c("Talked to Someone","Got Mad", "Avoidance","Problem Solving", "Worried/Cried"),
  class_labels = c("Adaptive Coping","Externalizing Behavior","No Coping"),
  class_legend_order = c(2,1,3),
  plot_title = "LCA Posterior Probability Plot"
  )
```



```
ggsave(here("figures", "C3_LCA_Rplot.png"), dpi=300, height=5, width=7, units="in")
```

Distal outcome plot

Note: The distal means are estimated with the binary covariate (Social Stress) fixed at the weighted average. This is specified by centering Social Stress as shown in the `Step-3` model syntax.

Read in Step3 model & extract model parameters

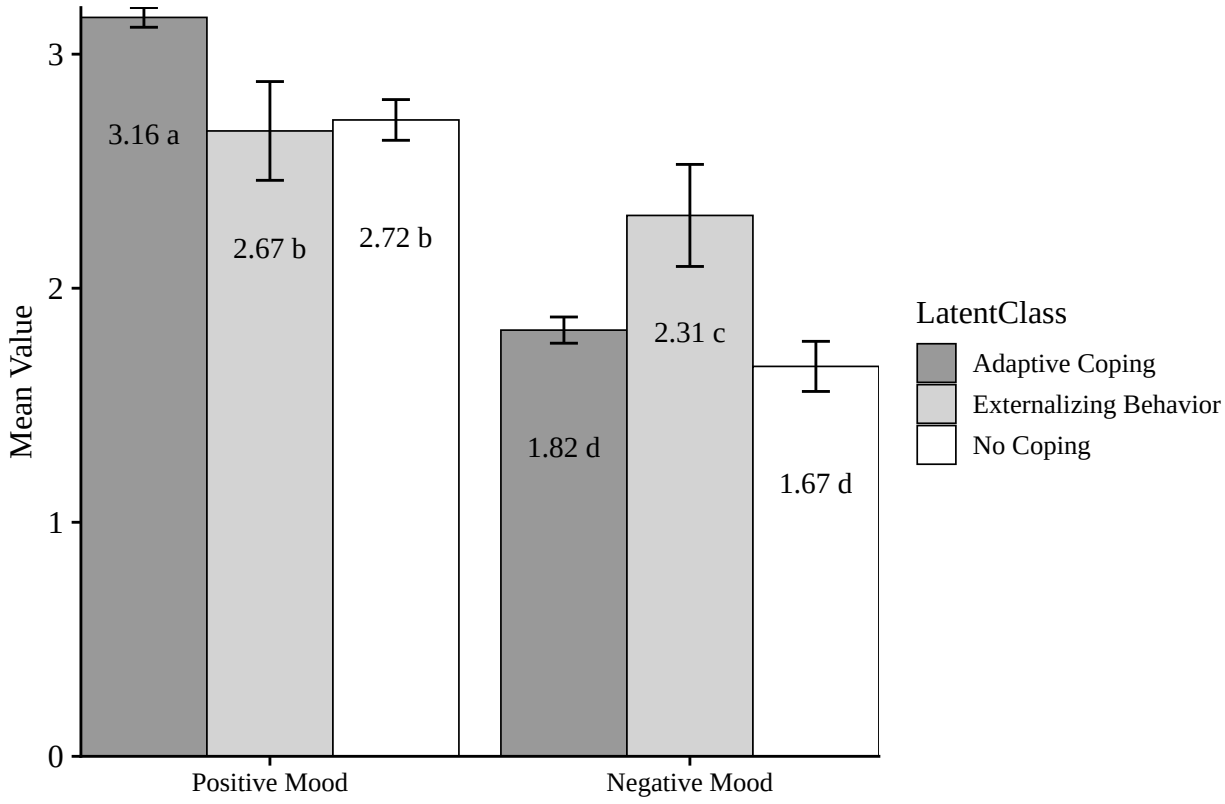
```
model_step3 <- readModels(here("3step_mplus", "Step3_3step_BD.out"), quiet = TRUE)

model_step3 <- data.frame(model_step3$parameters$unstandardized)
```

Create data-frame for distal outcome bar plot

```
distal_data <- model_step3 %>%
  filter(paramHeader == "Intercepts") %>%
  mutate(param = case_when(
      param == "POSMOOD1" ~ "Positive Mood",
      param == "NEGMOOD1" ~ "Negative Mood")) %>%
  mutate(LatentClass = factor(LatentClass,
      labels = c("Adaptive Coping", "Externalizing Behavior","No Coping"))) %>%
  mutate(value_labels = c("3.16 a", "1.82 d", "2.67 b", "2.31 c", "2.72 b", "1.67 d"))
```

Plot distal outcomes grouped by class (*Figure 3*)

```
ggplot(distal_data,
       aes(fill=LatentClass, y=est, x=fct_rev(param))) +
  geom_bar(position="dodge", stat="identity", color="black", size=.3) +
  geom_errorbar( aes(ymin=est-se, ymax=est+se),
                 width=.2, position=position_dodge(.9)) +
  geom_text(aes(y = est -.5, label = value_labels),
            family="Times New Roman", size=4,
            position=position_dodge(.9)) +
  scale_fill_grey(start = 0.6, end = 1.0) +
  theme_cowplot() +
  xlab("") + ylab("Mean Value") +
  theme(text=element_text(family="Times New Roman", size=12),
        axis.text.x=element_text(size=10)) +
        coord_cartesian(expand = FALSE)
```




```
# save plot
ggsave(here("figures","v4_barplot_grouped_class_Rplot.png"),
            dpi=300, height=4, width=6, units="in")
```

Simple slope plots

Note: The un-centered distal intercepts represent the conditional means when the binary covariate is at its first level `SOC_STRS = 0` (i.e., No Social Stress). Therefore, the conditional mean for `SOC_STRS = 1` (i.e., Social Stress) can be calculated by adding the associated slope coefficient to the intercept.

Read in the un-centered model & extract relevant parameters

```
model_uncen <- readModels(here("3step_mplus", "Step3_uncentered_BD.out"), quiet = TRUE)

model_uncen <- data.frame(model_uncen$parameters$unstandardized)

slope_data <- model_uncen %>%
  filter(str_detect(paramHeader, 'ON|Inter')) %>%
  unite("param", paramHeader:param, remove = TRUE) %>%
  mutate(param = str_replace(param, "MOOD1.ON_SOC_STRS", "_COEF")) %>%
```

```
  mutate(param = str_remove_all(param, "Intercepts_|MOOD1")) %>%
  mutate(LatentClass = factor(LatentClass,
    labels = c("Adaptive Coping (63.6%)", "Externalizing Behavior (5.3%)","No Coping (31.1%)")))
```

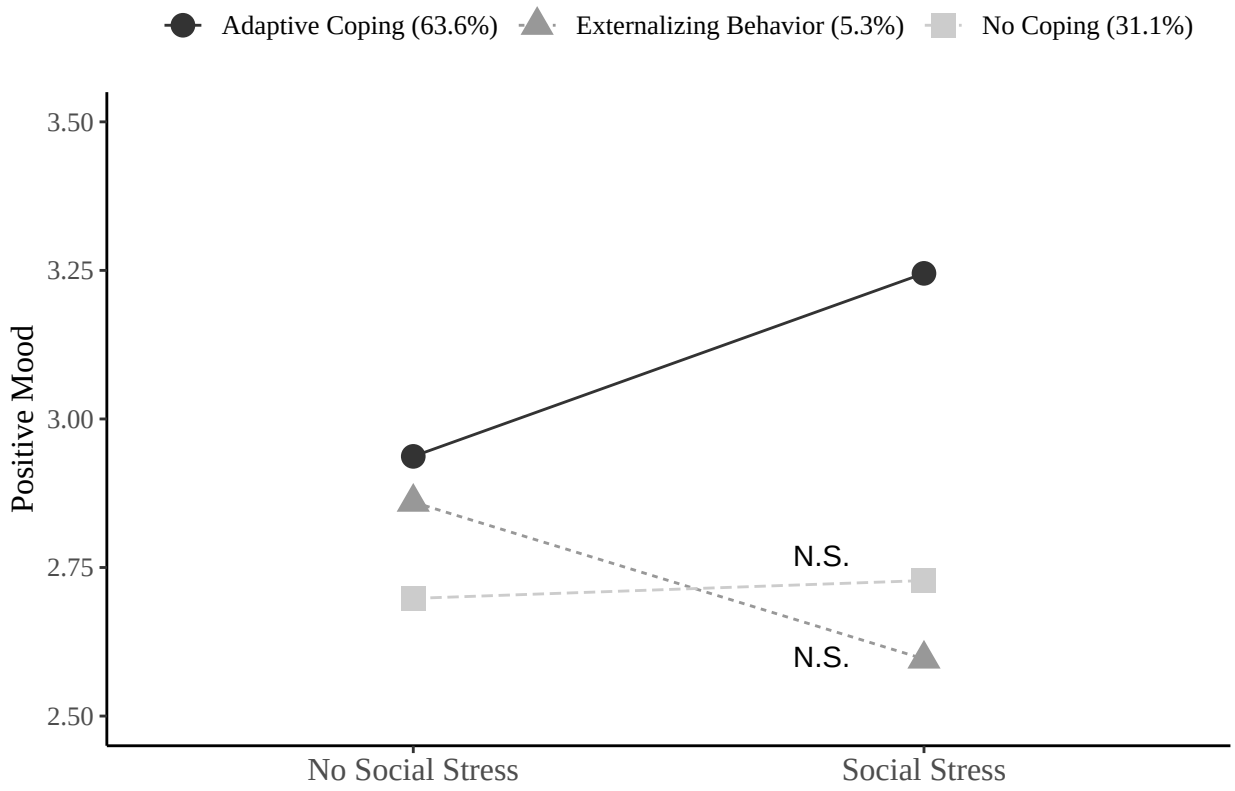
Positive mood simple slope graph

Prepare `data-frame` for plotting

```
pos_data <- slope_data %>%
  filter(str_detect(param, 'POS'))

pos_wide <- pos_data %>%
  select(param,est, LatentClass) %>%
  pivot_wider(names_from = param, values_from = est) %>%
  rename("No.Social.Stress" = 'POS') %>%
  mutate(Social.Stress = No.Social.Stress + POS_COEF) %>% # calculate conditional means for `SOC_STRS =
  select(-POS_COEF)

plot_pos <- melt(pos_wide, id.vars = "LatentClass") %>%
  mutate(variable = factor(variable,
                       levels = c("No.Social.Stress","Social.Stress"),
                       labels = c("No Social Stress","Social Stress")))
```

Plot positive mood simple slope graph

```
p_plot <- ggplot(plot_pos,
            aes(y=value, x=variable,
                color=LatentClass,
                group=LatentClass,
                shape=LatentClass,
                lty=LatentClass)) +
  geom_point(size = 4) + geom_line() +
  xlab("") + ylab("Positive Mood")  + ylim(2.5,3.5)+
  scale_colour_grey() +
  theme_classic() +
  theme(text=element_text(family="Times New Roman", size=12),
        axis.text.x=element_text(size=12),
        legend.text = element_text(family="Times New Roman", size=10),
        legend.position = "top", legend.title = element_blank()) +
      annotate(geom = "text",
           x = 1.8, y = 2.77,
           label = "N.S.", color = "black") +
      annotate(geom = "text",
           x = 1.8, y = 2.60,
           label = "N.S.", color = "black")

p_plot
```



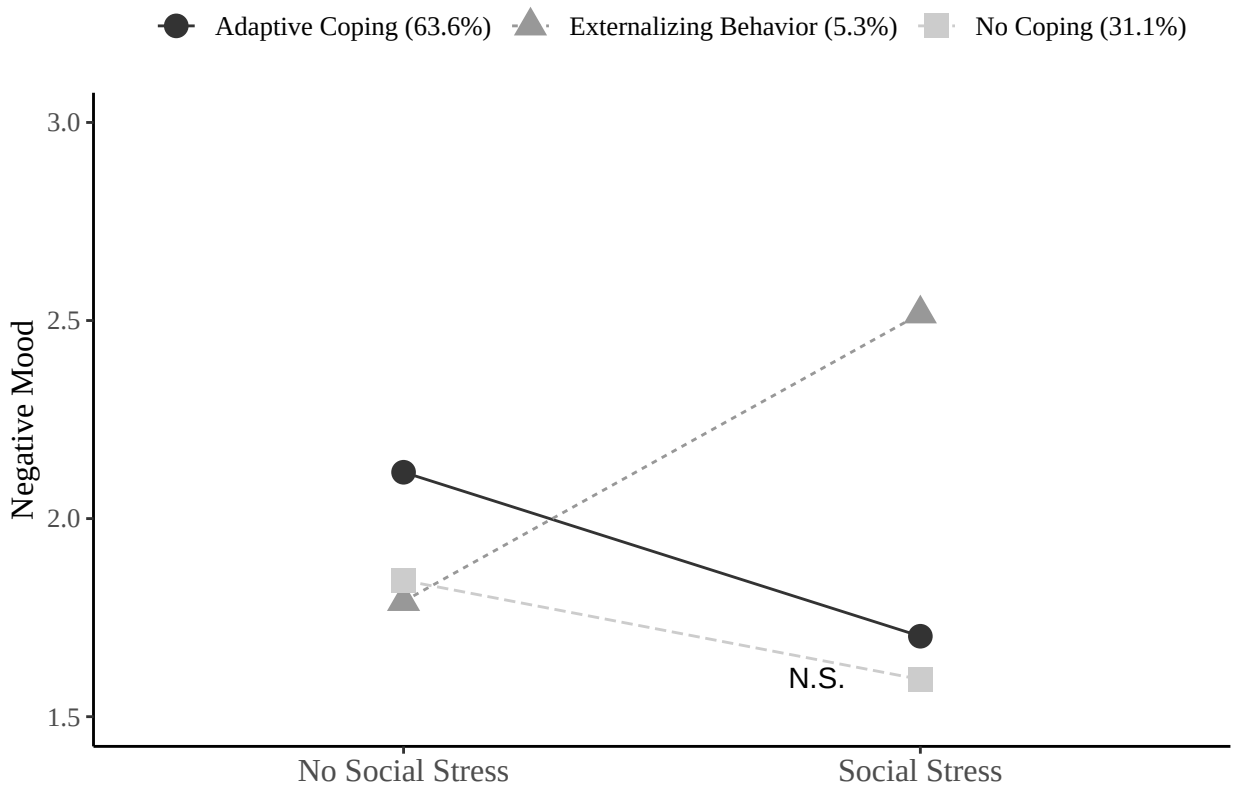
Negative mood simple slope graph

Prepare `data-frame` for plotting

```
neg_data <- slope_data %>%
  filter(str_detect(param, 'NEG'))

neg_wide <- neg_data %>%
  select(param,est, LatentClass) %>%
  pivot_wider(names_from = param, values_from = est) %>%
  rename("No.Social.Stress" = 'NEG') %>%
  mutate(Social.Stress = No.Social.Stress + NEG_COEF) %>% # calculate conditional means for `SOC_STRS =
  select(-NEG_COEF)

plot_neg <- melt(neg_wide, id.vars = "LatentClass") %>%
  mutate(variable = factor(variable,
                       levels = c("No.Social.Stress","Social.Stress"),
                       labels = c("No Social Stress","Social Stress")))
```

Plot negative mood simple slope graph

```r
n_plot <- ggplot(plot_neg,
            aes(y=value, x=variable,
            color=LatentClass,
            group=LatentClass,
            shape=LatentClass,
            lty=LatentClass)) +
  geom_point(size=4) + geom_line() +
  xlab("") + ylab("Negative Mood") + ylim(1.5,3)+
  scale_colour_grey() + theme_classic() +
  theme(text=element_text(family="Times New Roman", color = "black", size=12),
        axis.text.x=element_text(size=12),
        legend.text = element_text(family="Times New Roman", size=10),
        legend.position = "top", legend.title = element_blank()) +
    annotate(geom = "text",
       x = 1.8, y = 1.6,
       label = "N.S.", color = "black")

#view plot
n_plot
```



References:

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.