ABCD Multiview Step 1. Dataset, Simulation Topic, Questions, and Relevant References

Aidan Neher

## Motivating Questions

1. Motivating application: Which early life adversity (ELA), neuroimaging, and genomic features are important for the prediction of cognitive outcomes?
2. Methods extension: Does extension of the BIP framework to a mixed outcome model that accommodates observational clustering perform well, specifically above and beyond the BIP framework that does not account for observational clustering, in this setting and related settings?

## ABCD Study Dataset

The Adolescent Brain and Cognitive Development Study (ABCD Study®) is the largest long-term study of brain development and child health in the United States. Starting with students aged 9-10 years old and repeated until age 19-20, brains scans are taken bi-annually, biosamples, paper and pen tests, and iPad tasks annually, and interviews every 3-6 months.

We are most interested in the following “views”

* Early life adversity (ELA)
* Neuroimaging
  + sMRI: Cortical Thickness (CT) and Surface Area (SA)
  + fMRI: Functional Connectivity (FC)
* Genomics/ epigenomics

We are performing variable selection jointly with prediction in relation to the following continuous outcomes.

* internalizing problems,
* externalizing problems,
* cognitive flexibility, and
* inhibitory control

We consider the first two internalizing problems and externalizing problems primary outcomes and cognitive flexibility and inhibitory control secondary outcomes since the former are seemingly clinical outcomes and the latter are explanatory of the primary outcomes (Brieant et al. 2023).

We control for common covariates \* sex \* age \* race \* …

### Early Life Adversity (ELA)

Firstly, we load the ELA view using code shared by Mark Fiecas. Here is the additional context he provided on sharing: “This came from the supplementary material of the Orendain et al paper. Note that they actually used more, but these were the ones that were”relevant” per their factor analysis. They had about ~10 more variables but those did not have high factor loadings. Please see their supplemental material if you’d like to get the other ~10. You’ll need to revise Ellery’s standard prep code to make sure these get loaded in (see her “files” variable), and the first place to start is here, which is at the very bottom.”

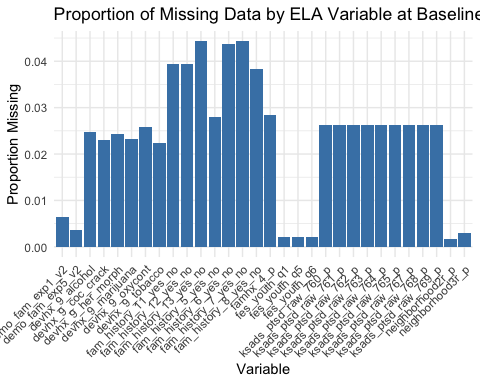
Let’s look at our ELA data.

str(ela\_data)

tibble [11,868 × 32] (S3: tbl\_df/tbl/data.frame)  
 $ src\_subject\_id : chr [1:11868] "NDAR\_INV003RTV85" "NDAR\_INV005V6D2C" "NDAR\_INV007W6H7B" "NDAR\_INV00BD7VDC" ...  
 $ eventname : chr [1:11868] "baseline\_year\_1\_arm\_1" "baseline\_year\_1\_arm\_1" "baseline\_year\_1\_arm\_1" "baseline\_year\_1\_arm\_1" ...  
 $ ksads\_ptsd\_raw\_761\_p : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ksads\_ptsd\_raw\_762\_p : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ksads\_ptsd\_raw\_763\_p : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ksads\_ptsd\_raw\_767\_p : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ksads\_ptsd\_raw\_768\_p : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ksads\_ptsd\_raw\_769\_p : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ksads\_ptsd\_raw\_760\_p : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ksads\_ptsd\_raw\_764\_p : num [1:11868] 0 0 0 0 1 0 0 0 0 0 ...  
 $ ksads\_ptsd\_raw\_765\_p : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ famhx\_4\_p : num [1:11868] 1 0 0 1 1 0 0 1 1 0 ...  
 $ fam\_history\_5\_yes\_no : num [1:11868] 0 0 0 0 1 0 0 0 0 0 ...  
 $ fam\_history\_6\_yes\_no : num [1:11868] 1 0 1 1 1 0 1 0 0 1 ...  
 $ fam\_history\_7\_yes\_no : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ fam\_history\_8\_yes\_no : num [1:11868] 0 0 0 0 0 0 0 0 1 0 ...  
 $ fam\_history\_11\_yes\_no: num [1:11868] 1 0 1 1 1 0 1 1 1 1 ...  
 $ fam\_history\_12\_yes\_no: num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ fam\_history\_13\_yes\_no: num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ neighborhood3r\_p : num [1:11868] 5 5 4 5 3 2 5 4 3 4 ...  
 $ neighborhood2r\_p : num [1:11868] 5 5 5 5 4 3 5 4 3 5 ...  
 $ devhx\_9\_tobacco : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ devhx\_9\_alcohol : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ devhx\_9\_marijuana : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ devhx\_9\_coc\_crack : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ devhx\_9\_her\_morph : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ devhx\_9\_oxycont : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ demo\_fam\_exp1\_v2 : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ demo\_fam\_exp5\_v2 : num [1:11868] 0 0 0 0 0 0 0 0 0 1 ...  
 $ fes\_youth\_q6 : num [1:11868] 0 0 0 0 0 0 0 0 0 0 ...  
 $ fes\_youth\_q1 : num [1:11868] 1 0 0 0 0 0 0 0 0 0 ...  
 $ fes\_youth\_q5 : num [1:11868] 1 0 0 0 0 0 0 0 0 0 ...

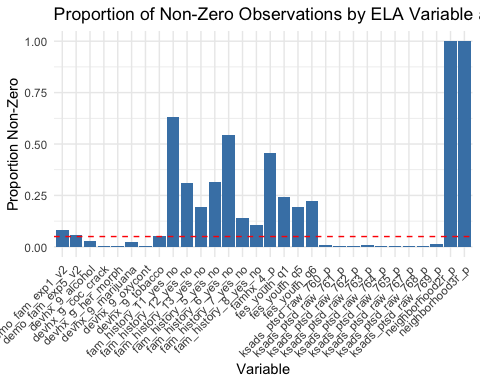
We check out the proportion of missingness in our ELA data.

# Calculate the proportion of missing values per column, excluding specific columns  
missing\_data <- ela\_data %>%  
 select(-src\_subject\_id, -eventname) %>% # Exclude these columns from the analysis  
 summarise(across(everything(), ~sum(is.na(.))/n())) %>%  
 pivot\_longer(cols = everything(), names\_to = "variable", values\_to = "prop\_missing")  
  
# Plot the proportion of missing data  
ggplot(missing\_data, aes(x = variable, y = prop\_missing)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 labs(x = "Variable", y = "Proportion Missing", title = "Proportion of Missing Data by ELA Variable at Baseline")



On removing NAs, we consider each variable’s the event rate (proportion of observations that are non-zero). We do this because we do not want to include variables with limited information/ possibly zero variance where all reported values are zero (i.e. ELA feature not observed in any observations).

# Calculate the proportion of non-zero values per column, excluding specific columns  
non\_zero\_data <- ela\_data %>%  
 select(-src\_subject\_id, -eventname) %>% # Exclude these columns from the analysis  
 summarise(across(everything(), ~mean(. != 0, na.rm = TRUE))) %>%  
 pivot\_longer(cols = everything(), names\_to = "variable", values\_to = "prop\_non\_zero")  
  
# Plot the proportion of non-zero data  
ggplot(non\_zero\_data, aes(x = variable, y = prop\_non\_zero)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 geom\_hline(yintercept = 0.05, linetype = "dashed", color = "red") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 labs(x = "Variable", y = "Proportion Non-Zero",   
 title = "Proportion of Non-Zero Observations by ELA Variable at Baseline")



print(paste("N Variables with zero variance:", sum(non\_zero\_data$prop\_non\_zero == 0)))

[1] "N Variables with zero variance: 0"

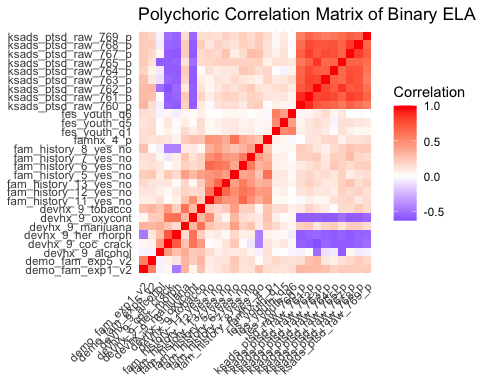
We notice a number of these variables included in the Orendain et al paper do not have event rates larger than 0.05, which is a quality control step implemented in the Brieant et al. 2023 paper.

We consider relationships between the ELA features by looking at polychoric correlations and TSNE plots. Note, for the polychoric correlation analysis, we filter to complete cases and also exclude the variables with five levels to only explore the binary variables (for now). In those with five levels, 1 = Strongly Disagree; 2 = Disagree; 3 = Neutral (neither agree nor disagree); 4 = Agree; 5 = Strongly Agree. It’s not clear how best to binarize/ handle these 5-level variables in the context of polychoric correlation calculation.

# Prepare the data by excluding non-relevant columns and ensuring complete cases  
filtered\_data <- ela\_data %>%  
 select(-src\_subject\_id, -eventname) %>%  
 na.omit()  
  
# Filter out variables with more than 2 unique levels  
binary\_data <- filtered\_data %>%  
 select\_if(~length(unique(.)) == 2)  
  
# Compute the polychoric correlation, only for binary variables  
polychoric\_matrix <- polychoric(binary\_data)$rho

Warning in cor.smooth(mat): Matrix was not positive definite, smoothing was  
done

# Transform the correlation matrix for visualization  
correlation\_data <- as.data.frame(polychoric\_matrix) %>%  
 rownames\_to\_column("Variable1") %>%  
 pivot\_longer(cols = -Variable1, names\_to = "Variable2", values\_to = "Correlation")  
  
# Plot the correlation matrix as a heatmap  
ggplot(correlation\_data, aes(x = Variable1, y = Variable2, fill = Correlation)) +  
 geom\_tile() +  
 scale\_fill\_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1)) +  
 labs(fill = "Correlation", title = "Polychoric Correlation Matrix of Binary ELA Variables", x = NULL, y = NULL)



# Organize variable pairs to ensure unique combinations  
high\_correlation\_pairs <- correlation\_data %>%  
 mutate(VariablePair = pmap\_chr(list(Variable1, Variable2), ~paste(sort(c(...)), collapse = "-"))) %>%  
 distinct(VariablePair, .keep\_all = TRUE) %>%  
 filter(abs(Correlation) > 0.75, Variable1 != Variable2)  
  
# View the table of feature pairs "VariablePair" with high correlations  
high\_correlation\_pairs %>%  
 select(VariablePair, Correlation) %>%  
 print

# A tibble: 10 × 2  
 VariablePair Correlation  
 <chr> <dbl>  
 1 ksads\_ptsd\_raw\_761\_p-ksads\_ptsd\_raw\_762\_p 0.787  
 2 ksads\_ptsd\_raw\_761\_p-ksads\_ptsd\_raw\_763\_p 0.808  
 3 ksads\_ptsd\_raw\_761\_p-ksads\_ptsd\_raw\_767\_p 0.815  
 4 ksads\_ptsd\_raw\_761\_p-ksads\_ptsd\_raw\_768\_p 0.775  
 5 ksads\_ptsd\_raw\_760\_p-ksads\_ptsd\_raw\_761\_p 0.884  
 6 ksads\_ptsd\_raw\_761\_p-ksads\_ptsd\_raw\_764\_p 0.775  
 7 ksads\_ptsd\_raw\_762\_p-ksads\_ptsd\_raw\_763\_p 0.786  
 8 ksads\_ptsd\_raw\_762\_p-ksads\_ptsd\_raw\_765\_p 0.852  
 9 ksads\_ptsd\_raw\_763\_p-ksads\_ptsd\_raw\_767\_p 0.768  
10 ksads\_ptsd\_raw\_767\_p-ksads\_ptsd\_raw\_768\_p 0.825

10 binary features included in Orendain et al paper’s analysis have high polychoric correlation. I’m wondering if the data processing in Brieant et al. 2023 paper is preferrable as a result in that they consolidated survey items that were highly correlated into composite scores.

Below is an exploratory tsne plot to demonstrate the heterogeneity in our sample’s ELA data at baseline. To ultimately generate this plot, we had to remove a large number of duplicates in our binary data.

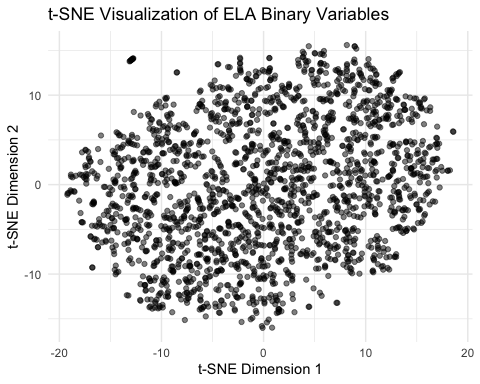
# Remove duplicate rows  
unique\_binary\_data <- binary\_data %>%   
 distinct()  
  
# Check the number of rows removed  
cat("Removed", nrow(binary\_data) - nrow(unique\_binary\_data), "duplicate rows.\n")

Removed 7900 duplicate rows.

# Set parameters for t-SNE  
set.seed(42) # for reproducibility  
tsne\_results <- Rtsne(as.matrix(unique\_binary\_data), dims = 2, perplexity = 30, verbose = TRUE, max\_iter = 500)

Performing PCA  
Read the 2071 x 28 data matrix successfully!  
Using no\_dims = 2, perplexity = 30.000000, and theta = 0.500000  
Computing input similarities...  
Building tree...  
Done in 0.29 seconds (sparsity = 0.056836)!  
Learning embedding...  
Iteration 50: error is 78.992256 (50 iterations in 0.24 seconds)  
Iteration 100: error is 78.992256 (50 iterations in 0.27 seconds)  
Iteration 150: error is 78.992256 (50 iterations in 0.38 seconds)  
Iteration 200: error is 78.992256 (50 iterations in 0.39 seconds)  
Iteration 250: error is 78.992256 (50 iterations in 0.47 seconds)  
Iteration 300: error is 4.097757 (50 iterations in 0.39 seconds)  
Iteration 350: error is 2.696579 (50 iterations in 0.22 seconds)  
Iteration 400: error is 2.588384 (50 iterations in 0.23 seconds)  
Iteration 450: error is 2.526801 (50 iterations in 0.23 seconds)  
Iteration 500: error is 2.490906 (50 iterations in 0.20 seconds)  
Fitting performed in 3.00 seconds.

# Create a data frame for plotting  
tsne\_data <- as.data.frame(tsne\_results$Y)  
colnames(tsne\_data) <- c("TSNE1", "TSNE2")  
  
# Plot t-SNE results  
ggplot(tsne\_data, aes(x = TSNE1, y = TSNE2)) +  
 geom\_point(alpha = 0.5) +  
 theme\_minimal() +  
 labs(title = "t-SNE Visualization of ELA Binary Variables",  
 x = "t-SNE Dimension 1",  
 y = "t-SNE Dimension 2")

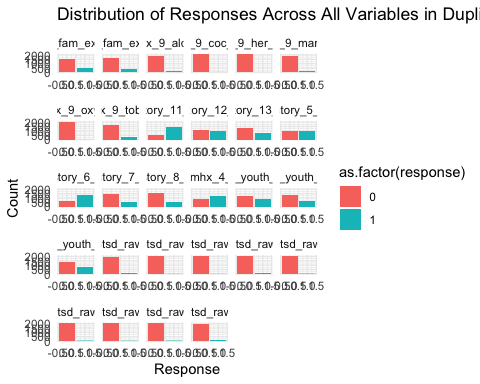


On attempting to make a tsne plot, we recognize that there are 7,900 duplicate rows in our binary data representative of 9,971 observations at baseline. This raises the question of what the characteristics are of these duplicates, are there specific classes of duplicates in our binary data e.g. the vast majority report no to all questions?

Note, 2,071 observations are included in the duplicate\_summary below, which means we have only 2,071 unique survey responses and 7,900 duplicate responses for a sample size in these binary baseline data of 9,071.

# Find duplicates (except for the first occurrence)  
duplicate\_data <- binary\_data %>%  
 add\_count() %>%   
 filter(n > 1) %>%  
 select(-n) %>%  
 distinct()  
  
# Count how many times each unique set of responses appears  
duplicate\_summary <- binary\_data %>%  
 group\_by(across(everything())) %>%  
 summarise(count = n(), .groups = 'drop') %>%  
 arrange(desc(count))  
  
# # Look at the most common duplicate entries  
# print(duplicate\_summary)  
  
# # For visualization, you may want to check specific questions or all questions  
# # Here's a general approach for visualizing the distribution of responses for one variable:  
# ggplot(duplicate\_summary, aes(x = ksads\_ptsd\_raw\_761\_p, y = count)) +  
# geom\_col() +  
# labs(title = "Distribution of Responses for a Specific Variable in Duplicate Rows",  
# x = "Response",  
# y = "Count of Duplicates")  
  
# Or visualize all variables' responses in duplicates  
duplicate\_summary\_long <- pivot\_longer(duplicate\_summary, cols = -count,   
 names\_to = "variable", values\_to = "response")  
  
ggplot(duplicate\_summary\_long, aes(x = response, fill = as.factor(response))) +  
 geom\_histogram(stat = "count") +  
 facet\_wrap(~ variable, scales = "free\_x") +  
 theme\_minimal() +  
 labs(title = "Distribution of Responses Across All Variables in Duplicate Rows",  
 x = "Response",  
 y = "Count")

Warning in geom\_histogram(stat = "count"): Ignoring unknown parameters:  
`binwidth`, `bins`, and `pad`



Features related to drug use and ptsd seem to be most absent in our duplicated binary observations.

#### Orendain et al. 2023 vs. Brieant et al. 2023

Noticing the discrepancy in the polychoric correlations allowed for between Orendain et al. 2023 (no polychoric correlation filtering/ feature composition), and Brieant et al. 2023 (features with polychoric correlation > 0.75 were transformed into composites to avoid potential collinearity concerns), we decide to compare and contrast the features included in each analysis.

In Orendain et al. 2023,

* “All adversity variables were binarized to indicate the presence or absence of exposure”. Did they binarize all features, including the neighborhood safety questions which has a scale from 1-5 available? If so, how did they do this?
* 17 additional features are reported in their Table S2 that were excluded from the final exploratory factor analysis because of a low factor loading (below 0.4). Might we include these? My initial thought is yes.

Authors included 47 adversity features of which 30 were found to be “important” in their exploratory factor analysis. These 30 features are those that have been characterized thus far in this report.

In Brieant et al. 2023,

Authors identify 139 ELA features that they then filter those with >50% missingness and <0.05% endorsement to pass 79 to a polychoric correlation matrix calculation. 26 were removed that had high correlation with other variables, and composites were added for a total of 60 variables to be used in the factor analysis/ Bayesian linear modeling.

We adapt code from Brieant et al. 2023’s Open Science Foundation (OSF) project 01\_data\_preparation/01\_pull\_variables.Rmd (https://osf.io/28cb7/?view\_only=b7789e2eb92d40d290358a5ad623ac65) to extract the 79 variables that passed the >50% missingness and <0.05% endorsement filtering steps. Note, seemingly authors count “src\_subject\_id” and “rel\_family\_id” as 2 of these 79 variables.

# Read in family variables  
rel <- read.csv(paste(data\_dir, 'abcd-general/abcd\_y\_lt.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, rel\_family\_id)  
  
# Read in family substance use summary scores  
fhx <- read.csv(paste(data\_dir, 'mental-health/mh\_p\_fhx.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, famhx\_ss\_fath\_prob\_alc\_p, famhx\_ss\_moth\_prob\_alc\_p, famhx\_ss\_fath\_prob\_dg\_p, famhx\_ss\_moth\_prob\_dg\_p)  
  
# Read in parent demographics  
pdemo <- read.csv(paste(data\_dir, 'abcd-general/abcd\_p\_demo.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, demo\_prim, demo\_prnt\_marital\_v2, demo\_prnt\_ed\_v2, demo\_prtnr\_ed\_v2, demo\_comb\_income\_v2,  
 demo\_fam\_exp1\_v2, demo\_fam\_exp2\_v2, demo\_fam\_exp3\_v2, demo\_fam\_exp4\_v2, demo\_fam\_exp5\_v2,   
 demo\_fam\_exp6\_v2, demo\_fam\_exp7\_v2)  
  
# Read in CRPBI  
crpbi <- read.csv(paste(data\_dir, 'culture-environment/ce\_y\_crpbi.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, crpbi\_parent1\_y, crpbi\_caregiver12\_y, crpbi\_parent2\_y, crpbi\_caregiver13\_y,  
 crpbi\_parent3\_y, crpbi\_caregiver14\_y, crpbi\_parent4\_y, crpbi\_caregiver15\_y, crpbi\_parent5\_y,   
 crpbi\_caregiver16\_y)  
  
# Read in parent report family environment scale  
fes02 <- read.csv(paste(data\_dir, 'culture-environment/ce\_p\_fes.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, fam\_enviro1\_p, fam\_enviro2r\_p, fam\_enviro3\_p, fam\_enviro4r\_p, fam\_enviro5\_p,  
 fam\_enviro6\_p, fam\_enviro7r\_p, fam\_enviro8\_p, fam\_enviro9r\_p)  
  
# Read in youth report family environment scale  
fes01 <- read.csv(paste(data\_dir, 'culture-environment/ce\_y\_fes.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, fes\_youth\_q1, fes\_youth\_q2, fes\_youth\_q3, fes\_youth\_q4, fes\_youth\_q5, fes\_youth\_q6,  
 fes\_youth\_q7, fes\_youth\_q8, fes\_youth\_q9)  
  
# Read in ksads trauma, parent interview  
ptsd <- read.csv(paste(data\_dir, 'mental-health/mh\_p\_ksads\_ptsd.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, ksads\_ptsd\_raw\_754\_p, ksads\_ptsd\_raw\_755\_p, ksads\_ptsd\_raw\_756\_p, ksads\_ptsd\_raw\_757\_p,  
 ksads\_ptsd\_raw\_758\_p, ksads\_ptsd\_raw\_759\_p, ksads\_ptsd\_raw\_760\_p, ksads\_ptsd\_raw\_761\_p,  
 ksads\_ptsd\_raw\_762\_p, ksads\_ptsd\_raw\_763\_p, ksads\_ptsd\_raw\_764\_p, ksads\_ptsd\_raw\_765\_p,  
 ksads\_ptsd\_raw\_766\_p, ksads\_ptsd\_raw\_767\_p, ksads\_ptsd\_raw\_768\_p, ksads\_ptsd\_raw\_769\_p,  
 ksads\_ptsd\_raw\_770\_p)  
  
# Read in parental monitoring  
pmq <- read.csv(paste(data\_dir, 'culture-environment/ce\_y\_pm.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, parent\_monitor\_q1\_y, parent\_monitor\_q2\_y, parent\_monitor\_q3\_y, parent\_monitor\_q4\_y,  
 parent\_monitor\_q5\_y)  
  
# Read in neighborhood safety and crime, parents  
pnscss <- read.csv(paste(data\_dir, 'culture-environment/ce\_p\_nsc.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, nsc\_p\_ss\_mean\_3\_items)  
  
# Read in neighborhood safety and crime, youth  
ynsc <- read.csv(paste(data\_dir, 'culture-environment/ce\_y\_nsc.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, neighborhood\_crime\_y)  
  
# Read in ASR (parent psychopathology)  
asr <- read.csv(paste(data\_dir, 'mental-health/mh\_p\_asr.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, asr\_scr\_anxdisord\_r, asr\_scr\_somaticpr\_r, asr\_scr\_depress\_r, asr\_scr\_avoidant\_r,  
 asr\_scr\_adhd\_r, asr\_scr\_antisocial\_r, asr\_scr\_inattention\_r, asr\_scr\_hyperactive\_r)  
  
# Read in ADI data file  
ADI <- read.csv(paste(data\_dir, 'linked-external-data/led\_l\_adi.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, reshist\_addr1\_adi\_wsum)  
  
# Merge all data frames  
ela\_data\_brieant <- full\_join(fhx, asr) %>%  
 full\_join(pdemo) %>%  
 full\_join(crpbi) %>%  
 full\_join(fes01) %>%  
 full\_join(fes02) %>%  
 full\_join(ptsd) %>%  
 full\_join(pmq) %>%  
 full\_join(pnscss) %>%  
 full\_join(ynsc) %>%  
 full\_join(rel) %>%  
 full\_join(ADI)

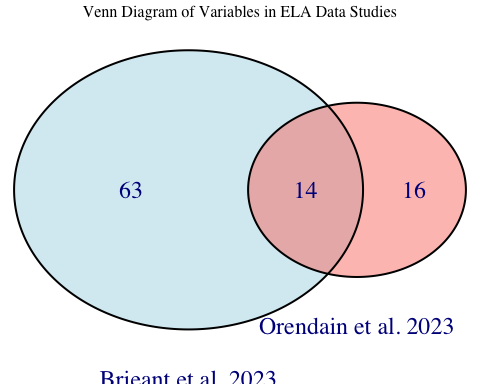
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`  
Joining with `by = join\_by(src\_subject\_id)`

print(paste("ncols of ELA data used for Brieant el al. 2023 analysis:",  
 ncol(ela\_data\_brieant)))

[1] "ncols of ELA data used for Brieant el al. 2023 analysis: 79"

We see the extent to which the variables included by Brient and colleagues overlap with the 60 considered important by Orendain et al. 2023.

# Exclude specified variables and extract variable names from each dataset  
vars\_ela\_orendain <- setdiff(names(ela\_data),   
 c("src\_subject\_id", "eventname", "rel\_family\_id"))  
vars\_ela\_brieant <- setdiff(names(ela\_data\_brieant),   
 c("src\_subject\_id", "eventname", "rel\_family\_id"))  
  
# Draw a Venn diagram  
venn.plot <- venn.diagram(  
 x = list(  
 Orendain = vars\_ela\_orendain,  
 Brieant = vars\_ela\_brieant  
 ),  
 filename = NULL, # Set to NULL for plotting directly in R  
 category.names = c("Orendain et al. 2023", "Brieant et al. 2023"),  
 output = TRUE,  
 main = "Venn Diagram of Variables in ELA Data Studies",  
 fill = c("salmon", "lightblue"),  
 alpha = 0.50,  
 label.col = "darkblue",  
 cex = 1.5,  
 fontfamily = "serif",  
 cat.cex = 1.5,  
 cat.col = "darkblue",  
 cat.pos = 0,  
 cat.dist = 0.07,  
 cat.fontfamily = "serif"  
)  
  
# Display the Venn diagram  
grid.draw(venn.plot)



We discover these studies chose quite different sets of ELA variables for their analyses with only 14 shared between the two. This makes me wonder two things,

1. Would the amount of overlap be different had we included variables from upstream in these authors respective data processing and analysis pipelines?
2. What accounts for the limited overlap in the variables used for each analysis? Could it be that variables included by Orendain et al. 2023 would have been excluded by Brieant et al. 2023’s missingness and <0.05% endorsement filters? What explains the 63 variables included by strictly Brieant et al. 2023? Is subjective judgement to blame, or are there other justifications available?

### Covariates and Outcomes

Now we load the covariate and outcome data, adapting code from Ellery’s “ABCD\_data\_prep.Rmd”. Note, in user input chunk, you can change the data directory and the output directory. Also, see the ABCD data dictionary (https://data-dict.abcdstudy.org/?), or the online codebook (https://docs.google.com/spreadsheets/d/1uHRrXASaxtZbRAeJvPJktqxyDkPdA2\_WgxYdLTjhK1Q/edit?usp=sharing) to learn more about the variable naming.

# For Ellery's code, we must specify what time points we want.   
# Put the string provided for all desired timepoints into a vector e.g. c('timepoint1\_name','timepoint2\_name') etc.  
timepoint\_list <- c("baseline\_year\_1\_arm\_1")

Load Data Files

# Function to load the required datasets  
load\_datasets <- function(data\_dir) {  
 # Standard files for most studies  
 demog <- read\_csv(file.path(data\_dir, "abcd-general/abcd\_p\_demo.csv"))  
 demog\_y <- read\_csv(file.path(data\_dir, "mental-health/mh\_y\_ksads\_bg.csv"))  
 puberty <- read\_csv(file.path(data\_dir, "physical-health/ph\_p\_pds.csv"))  
 study\_covars <- read\_csv(file.path(data\_dir, "abcd-general/abcd\_y\_lt.csv"))  
 sib\_twin <- read\_csv(file.path(data\_dir, "genetics/gen\_y\_pihat.csv"))  
   
 # MRI standard files  
 mri <- read\_csv(file.path(data\_dir, "imaging/mri\_y\_adm\_info.csv"))  
 qc <- read\_csv(file.path(data\_dir, "imaging/mri\_y\_qc\_incl.csv"))  
 scan\_qtns <- read\_csv(file.path(data\_dir, "imaging/mri\_y\_adm\_qtn.csv"))  
   
 # Standard files for clinical data  
 cbcl <- read\_csv(file.path(data\_dir, "mental-health/mh\_p\_cbcl.csv"))  
 bpm\_y <- read\_csv(file.path(data\_dir, "mental-health/mh\_y\_bpm.csv"))  
 ksad\_p <- read\_csv(file.path(data\_dir, "mental-health/mh\_p\_ksads\_ss.csv"))  
 ksad\_y <- read\_csv(file.path(data\_dir, "mental-health/mh\_y\_ksads\_ss.csv"))  
   
 # Combine all datasets into a list (if needed for further processing or return)  
 datasets <- list(demog = demog, demog\_y = demog\_y, puberty = puberty,  
 study\_covars = study\_covars, sib\_twin = sib\_twin, mri = mri,  
 qc = qc, scan\_qtns = scan\_qtns, cbcl = cbcl, bpm\_y = bpm\_y,  
 ksad\_p = ksad\_p, ksad\_y = ksad\_y)  
   
 return(datasets)  
}  
  
# Load the datasets  
datasets <- load\_datasets(data\_dir)

Warning: One or more parsing issues, call `problems()` on your data frame for details,  
e.g.:  
 dat <- vroom(...)  
 problems(dat)

Rows: 48807 Columns: 276  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (2): src\_subject\_id, eventname  
dbl (268): demoi\_p\_select\_language\_\_\_1, demo\_prim, demo\_brthdat\_v2, demo\_ed\_...  
lgl (6): demo\_adopt\_agex\_v2\_bl\_dk, demo\_years\_us\_v2, fam\_roster\_12c\_v2\_l, ...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
Rows: 49012 Columns: 62  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (2): src\_subject\_id, eventname  
dbl (60): kbi\_y\_grade\_repeat, kbi\_y\_drop\_in\_grades, kbi\_y\_det\_susp, kbi\_y\_de...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
Rows: 49151 Columns: 32  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (2): src\_subject\_id, eventname  
dbl (29): pds\_select\_language\_\_\_1, pubertal\_sex\_p, pds\_1\_p, pds\_2\_p, pds\_3\_...  
date (1): menstrualcycle1\_p  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Warning: One or more parsing issues, call `problems()` on your data frame for details,  
e.g.:  
 dat <- vroom(...)  
 problems(dat)

Rows: 90312 Columns: 10  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (4): src\_subject\_id, eventname, site\_id\_l, interview\_date  
dbl (6): rel\_family\_id, rel\_birth\_id, school\_id, district\_id, interview\_age,...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Warning: One or more parsing issues, call `problems()` on your data frame for details,  
e.g.:  
 dat <- vroom(...)  
 problems(dat)

Rows: 11868 Columns: 52  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (5): src\_subject\_id, eventname, genetic\_paired\_subjectid\_1, genetic\_pai...  
dbl (44): rel\_family\_id, rel\_birth\_id, rel\_group\_id, rel\_ingroup\_order, rel\_...  
lgl (3): genetic\_pi\_hat\_4, genetic\_zygosity\_status\_4, genetic\_paired\_subjec...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
Rows: 22939 Columns: 9  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (7): src\_subject\_id, eventname, mri\_info\_visitid, mri\_info\_manufacturer,...  
dbl (2): mri\_info\_magneticfieldstrength, mri\_info\_studydate  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
Rows: 22939 Columns: 9  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (2): src\_subject\_id, eventname  
dbl (7): imgincl\_t1w\_include, imgincl\_t2w\_include, imgincl\_dmri\_include, img...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
Rows: 23320 Columns: 42  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (2): src\_subject\_id, eventname  
dbl (40): prescan\_relaxed\_1, prescan\_happy\_1, prescan\_scared\_1, prescan\_awak...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Warning: One or more parsing issues, call `problems()` on your data frame for details,  
e.g.:  
 dat <- vroom(...)  
 problems(dat)

Rows: 48737 Columns: 203  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (2): src\_subject\_id, eventname  
dbl (180): cbcl\_select\_language\_\_\_1, cbcl\_q01\_p, cbcl\_q02\_p, cbcl\_q03\_p, cbc...  
lgl (21): cbcl\_scr\_syn\_anxdep\_m, cbcl\_scr\_syn\_withdep\_m, cbcl\_scr\_syn\_somat...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
Rows: 78578 Columns: 50  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (2): src\_subject\_id, eventname  
dbl (48): bpm\_1\_y, bpm\_2\_y, bpm\_3\_y, bpm\_4\_y, bpm\_5\_y, bpm\_6\_y, bpm\_7\_y, bpm...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
Rows: 48690 Columns: 1911  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (6): src\_subject\_id, eventname, ksads\_import\_id\_p, ksads\_timestamp\_p,...  
dbl (1902): ksads\_1\_843\_p, ksads\_1\_845\_p, ksads\_1\_844\_p, ksads\_1\_840\_p, ksad...  
lgl (3): ksads2\_13\_49\_p, ksads2\_20\_19\_p, ksads2\_comments\_p  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
Rows: 48790 Columns: 1912  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (6): src\_subject\_id, eventname, ksads\_import\_id\_t, ksads\_timestamp\_t,...  
dbl (1903): ksads\_1\_1\_t, ksads\_1\_2\_t, ksads\_1\_3\_t, ksads\_1\_4\_t, ksads\_1\_5\_t,...  
lgl (3): ksads2\_20\_19\_t, ksads2\_23\_99\_t, ksads2\_comments\_t  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Make data.frames accessible to global environment   
attach(datasets)

# KSAD INITIAL CLEANING  
# Why? During the data collection, a new version of the KSAD was used (KSAD2), so each KSAD variable has a corresponding KSAD2 variable which contains the data since the new version was adopted. The code below combines these variables into one.  
  
ksad\_new\_df <- ksad\_p %>% # initialize new data frame  
 select(src\_subject\_id, eventname)  
   
clean\_ksads <- function(ksad\_new\_df, ksad, ksad2, p\_or\_t){   
 # goal: combine two ksad variables into one new var  
 # ksad\_new\_df = a new data frame created above to hold the new vars,   
 # ksad = 1st ksad variable (old ksad),   
 # ksad2 = ksad2 var (same content/question as ksad1),   
 # p\_or\_t = whether the variable is a p or t ksad   
 # output: a dataframe containing the new variable  
 assign("ksad", paste0(ksad, sep = "\_", p\_or\_t)) # create ksad variable from inputs and assign it the name ksad  
 assign("ksad2", paste0(ksad2, sep = "\_", p\_or\_t)) # create ksad2 variable from inputs and assign it the name ksad2  
 assign("ksad\_new", paste0(ksad, "\_new")) # create new variable name from inputs and assign it the name ksad\_new  
 assign("ksad\_df", paste("ksad", if\_else(p\_or\_t == "p", "p", "y"), sep = "\_")) # create data frame name from inputs and name it ksad\_df  
 my\_cols <- c("src\_subject\_id", "eventname", ksad, ksad2) # create vector of necessary column names  
 intermediary <- get(ksad\_df) %>%  
 select(all\_of(my\_cols)) %>% # select cols (need to do it this way because I'm referring to the cols as strings)  
 mutate(!!ksad\_new := if\_else(is.na(get(ksad)) == T, get(ksad2), get(ksad))) # create new var  
 ksad\_new\_df <- right\_join(intermediary, ksad\_new\_df, by = c("src\_subject\_id", "eventname")) # join new var to existing data set  
 ksad\_new\_df <- ksad\_new\_df %>%  
 select(src\_subject\_id, eventname, ends\_with("\_new")) # select only necessary vars  
 return(ksad\_new\_df)  
}  
  
  
# call the function with each pair of ksad and ksad2 variables (make sure to assign the output to the same name so you create one data set with all the new variables)  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_946", "ksads2\_23\_906", "p") # suicidal ideation  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_947", "ksads2\_23\_907", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_948", "ksads2\_23\_908", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_949", "ksads2\_23\_909", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_950", "ksads2\_23\_910", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_951", "ksads2\_23\_911", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_957", "ksads2\_23\_917", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_958", "ksads2\_23\_918", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_959", "ksads2\_23\_919", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_960", "ksads2\_23\_920", "p")   
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_961", "ksads2\_23\_921", "p")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_954", "ksads2\_23\_914", "p") # suicidal attempts, prep  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_965", "ksads2\_23\_925", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_962", "ksads2\_23\_922", "p")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_966", "ksads2\_23\_926", "p") # NO suicidal ideation/behaviors  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_955", "ksads2\_23\_915", "p")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_143", "ksads2\_23\_134", "p") # non-suicidal self injury  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_144", "ksads2\_23\_135", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_956", "ksads2\_23\_916", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_945", "ksads2\_23\_905", "p")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_840", "ksads2\_1\_790", "p") # depression  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_841", "ksads2\_1\_791", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_842", "ksads2\_1\_792", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_843", "ksads2\_1\_793", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_844", "ksads2\_1\_794", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_845", "ksads2\_1\_795", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_846", "ksads2\_1\_796", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_847", "ksads\_1\_847", "p") # this one does not have a corresponding ksad2 (I put it here so it would still appear in the df)  
  
# repeat function calls with t instead of p  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_946", "ksads2\_23\_906", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_947", "ksads2\_23\_907", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_948", "ksads2\_23\_908", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_949", "ksads2\_23\_909", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_950", "ksads2\_23\_910", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_957", "ksads2\_23\_917", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_958", "ksads2\_23\_918", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_959", "ksads2\_23\_919", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_960", "ksads2\_23\_920", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_961", "ksads2\_23\_921", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_954", "ksads2\_23\_914", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_965", "ksads2\_23\_925", "t")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_966", "ksads2\_23\_926", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_955", "ksads2\_23\_915", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_951", "ksads2\_23\_911", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_962", "ksads2\_23\_922", "t")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_143", "ksads2\_23\_134", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_144", "ksads2\_23\_135", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_956", "ksads2\_23\_916", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_945", "ksads2\_23\_905", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_963", "ksads2\_23\_923", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_964", "ksads2\_23\_924", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_953", "ksads2\_23\_913", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_952", "ksads2\_23\_912", "t")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_840", "ksads2\_1\_790", "t")   
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_841", "ksads2\_1\_791", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_842", "ksads2\_1\_792", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_843", "ksads2\_1\_793", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_844", "ksads2\_1\_794", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_845", "ksads2\_1\_795", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_846", "ksads2\_1\_796", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_847", "ksads\_1\_847", "t") # this one does not have a corresponding ksad2 (I put it here so it would still appear in the df)

Demographic Cleaning Pre-Merge

# COMBINE BASELINE VARIABLES WITH LONGITUDINAL VARIABLES  
# how it combines: if the observation occurred in year1/arm1 (the baseline) the combination variable takes the value of the baseline variable, if not the combo variable takes the value of the longitudinal variable  
var\_vect <- c("demo\_brthdat\_v2", "demo\_gender\_id\_v2", "demo\_prnt\_ed\_v2", "demo\_prtnr\_ed\_v2" , "demo\_prnt\_marital\_v2", "demo\_comb\_income\_v2", "demo\_roster\_v2", "demo\_fam\_exp1\_v2", "demo\_fam\_exp2\_v2", "demo\_fam\_exp3\_v2", "demo\_fam\_exp4\_v2", "demo\_fam\_exp5\_v2", "demo\_fam\_exp6\_v2", "demo\_fam\_exp7\_v2") # input any baseline variable name you wish to combine with long. var in this vector  
combine.vars <- function(var){ # takes baseline variable name (a string) and makes a new variable using the corresponding long. var, outputs demog dataset with new combo variables (combo variables will be named the baseline variable name with "\_comb" at the end)  
 assign("var\_comb", paste0(var, "\_comb"))  
 assign("var\_l", paste0(var, "\_l"))  
 demog <- demog %>%  
 mutate(!!var\_comb := if\_else(eventname == "baseline\_year\_1\_arm\_1", get(var), get(var\_l))) # if observation occurred in year 1, arm 1 use bl var if not use long var  
 return(demog)  
}  
  
for (i in 1:length(var\_vect)) { # loops through vector of all variables to be combined and applies the combine.vars function  
 demog <- combine.vars(var\_vect[i])}

Select Relevant Variables

demog\_bl <- demog[demog$eventname == "baseline\_year\_1\_arm\_1",]  
  
demog\_bl <- demog\_bl %>%   
 select(src\_subject\_id, demo\_sex\_v2, demo\_race\_a\_p\_\_\_10,  
 demo\_race\_a\_p\_\_\_11,demo\_race\_a\_p\_\_\_12, demo\_race\_a\_p\_\_\_13, demo\_race\_a\_p\_\_\_14,  
 demo\_race\_a\_p\_\_\_15, demo\_race\_a\_p\_\_\_16, demo\_race\_a\_p\_\_\_17, demo\_race\_a\_p\_\_\_18,  
 demo\_race\_a\_p\_\_\_19, demo\_race\_a\_p\_\_\_20, demo\_race\_a\_p\_\_\_21, demo\_race\_a\_p\_\_\_22,  
 demo\_race\_a\_p\_\_\_23, demo\_race\_a\_p\_\_\_24, demo\_race\_a\_p\_\_\_25,demo\_race\_a\_p\_\_\_77,   
 demo\_race\_a\_p\_\_\_99, demo\_ethn\_v2, demo\_prnt\_marital\_v2, demo\_prnt\_ed\_v2, demo\_prtnr\_ed\_v2, demo\_comb\_income\_v2)  
  
demog\_bl <- dplyr::rename(demog\_bl, demo\_prnt\_marital\_v2\_bl = demo\_prnt\_marital\_v2)  
demog\_bl <- dplyr::rename(demog\_bl, demo\_comb\_income\_v2\_bl = demo\_comb\_income\_v2)  
demog\_bl <- dplyr::rename(demog\_bl, demo\_prnt\_ed\_v2\_bl = demo\_prnt\_ed\_v2)  
demog\_bl <- dplyr::rename(demog\_bl, demo\_prtnr\_ed\_v2\_bl = demo\_prtnr\_ed\_v2)  
  
demog <- demog %>%  
 select(src\_subject\_id, eventname, demo\_brthdat\_v2\_comb, demo\_gender\_id\_v2\_comb,   
 demo\_prnt\_ed\_v2\_comb, demo\_prtnr\_ed\_v2\_comb, demo\_prnt\_marital\_v2\_comb, demo\_comb\_income\_v2\_comb,  
 demo\_roster\_v2\_comb, demo\_fam\_exp1\_v2\_comb, demo\_fam\_exp2\_v2\_comb, demo\_fam\_exp3\_v2\_comb,   
 demo\_fam\_exp4\_v2\_comb, demo\_fam\_exp5\_v2\_comb, demo\_fam\_exp6\_v2\_comb, demo\_fam\_exp7\_v2\_comb,  
 acs\_raked\_propensity\_score)  
  
demog\_y <- demog\_y %>%  
 select(src\_subject\_id, eventname, kbi\_gender, kbi\_y\_trans\_id, kbi\_y\_sex\_orient)  
  
puberty <- puberty %>%  
 select(src\_subject\_id, eventname, pds\_p\_ss\_male\_category\_2, pds\_p\_ss\_female\_category\_2)  
  
fam <- study\_covars[study\_covars$eventname == "baseline\_year\_1\_arm\_1",]  
  
fam <- fam %>%  
 select(src\_subject\_id, rel\_family\_id)  
  
study\_covars <- study\_covars %>%  
 select(src\_subject\_id, eventname, site\_id\_l, interview\_age)  
  
sib\_twin <- sib\_twin %>%   
 select(src\_subject\_id, rel\_relationship, rel\_group\_id)  
  
mri <- mri %>%  
 select(src\_subject\_id, eventname, mri\_info\_manufacturer)  
  
qc <- qc %>%  
 select(src\_subject\_id, eventname, imgincl\_rsfmri\_include)  
  
cbcl <- cbcl %>%  
 select(src\_subject\_id, eventname, cbcl\_scr\_syn\_internal\_r, cbcl\_scr\_syn\_external\_r, cbcl\_scr\_syn\_totprob\_r,  
 cbcl\_scr\_dsm5\_depress\_r, cbcl\_scr\_dsm5\_anxdisord\_r, cbcl\_scr\_dsm5\_adhd\_r,  
 cbcl\_scr\_syn\_internal\_t, cbcl\_scr\_syn\_external\_t, cbcl\_scr\_syn\_totprob\_t,  
 cbcl\_scr\_dsm5\_depress\_t, cbcl\_scr\_dsm5\_anxdisord\_t, cbcl\_scr\_dsm5\_adhd\_t)  
  
bpm\_y <- bpm\_y %>%   
 select(src\_subject\_id, eventname, bpm\_y\_scr\_attention\_r, bpm\_y\_scr\_attention\_t, bpm\_y\_scr\_internal\_r,   
 bpm\_y\_scr\_internal\_t, bpm\_y\_scr\_external\_r, bpm\_y\_scr\_external\_t, bpm\_y\_scr\_totalprob\_r,   
 bpm\_y\_scr\_totalprob\_t)

Merge into single file and replace missing data codes with NA (Can change this if you do not want that)

files <- list(demog, demog\_y, puberty, study\_covars,   
 mri, qc, scan\_qtns, cbcl, bpm\_y, ksad\_new\_df)  
abcd\_data\_0 <- files %>% reduce(full\_join, by = c("src\_subject\_id", "eventname"))  
  
files\_2 <- list(demog\_bl, fam, sib\_twin, abcd\_data\_0)  
abcd\_data <- files\_2 %>% reduce(full\_join, by = "src\_subject\_id")  
  
abcd\_data <- abcd\_data %>%  
 mutate(across(where(is.numeric), ~na\_if(.,777))) %>%  
 mutate(across(where(is.numeric), ~na\_if(.,999))) %>%  
 mutate(across(where(is.numeric), ~na\_if(.,555))) %>%  
 mutate(across(where(is.numeric), ~na\_if(.,888))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"777"))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"999"))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"555"))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"888"))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"")))

Clean and Recode Age

abcd\_data <- abcd\_data %>%  
 mutate(demo\_brthdat\_v2\_comb\_clean = if\_else(demo\_brthdat\_v2\_comb > 21,   
 demo\_brthdat\_v2\_comb/12, demo\_brthdat\_v2\_comb), # convert months to years  
 demo\_brthdat\_v2\_comb\_clean = if\_else(demo\_brthdat\_v2\_comb\_clean < 8, NA, demo\_brthdat\_v2\_comb\_clean), # younger than 8 --> NA  
 demo\_brthdat\_v2\_comb\_clean = trunc(demo\_brthdat\_v2\_comb\_clean)) %>% # remove decimals   
 mutate(interview\_age\_b = interview\_age / 12) # convert months to years

Recode Race Variables

abcd\_data <- abcd\_data %>%  
 mutate(demo\_ethn\_v2 = abs(demo\_ethn\_v2 - 2), # change "2" to 0 to match other vars   
 White\_race = demo\_race\_a\_p\_\_\_10,  
 Black\_race = demo\_race\_a\_p\_\_\_11,  
 AIAN\_race = if\_else(rowSums(select(., num\_range("demo\_race\_a\_p\_\_\_", 12:13)))  
 >=1, 1, 0),  
 NHPI\_race = if\_else(rowSums(select(., num\_range("demo\_race\_a\_p\_\_\_", 14:17)))  
 >=1, 1, 0),  
 Asian\_race = if\_else(rowSums(select(., num\_range("demo\_race\_a\_p\_\_\_", 18:24)))  
 >=1, 1, 0),  
 Other\_race = demo\_race\_a\_p\_\_\_25,  
 Missing\_race = if\_else(demo\_race\_a\_p\_\_\_99 == 1, 1, demo\_race\_a\_p\_\_\_77),   
 Missing\_race = if\_else(White\_race == 1 | # if participant did not endorse any of these, mark as missing  
 Black\_race == 1 |   
 AIAN\_race == 1 |  
 NHPI\_race == 1 |  
 Asian\_race == 1 |   
 Other\_race == 1 |  
 demo\_ethn\_v2 == 1, 0, 1),  
 Missing\_race = if\_else(is.na(Missing\_race) == T, 1, Missing\_race)) %>% # mark all NAs in missing race as 1  
 mutate(Indigenous\_race = if\_else(AIAN\_race == 1 | NHPI\_race == 1, 1, 0))

Recode Puberty Variable

abcd\_data <- abcd\_data %>%  
 mutate(pubertal\_status = if\_else(demo\_sex\_v2 == '1' | demo\_sex\_v2 == '3',  
 pds\_p\_ss\_male\_category\_2, # Set to this if the condition above is TRUE  
 pds\_p\_ss\_female\_category\_2)) # Otherwise set to this

Recode Parent-Related Demographics

abcd\_data <- abcd\_data %>%  
 mutate(across(c(demo\_prnt\_ed\_v2\_comb, demo\_prtnr\_ed\_v2\_comb,   
 demo\_prnt\_ed\_v2\_bl, demo\_prtnr\_ed\_v2\_bl), ~as.integer(.x))) %>%  
 mutate(highest\_demo\_ed\_comb = case\_when(  
 is.na(demo\_prtnr\_ed\_v2\_comb) == T ~   
 demo\_prnt\_ed\_v2\_comb,  
 demo\_prnt\_ed\_v2\_comb > demo\_prtnr\_ed\_v2\_comb ~   
 demo\_prnt\_ed\_v2\_comb,  
 demo\_prnt\_ed\_v2\_comb <= demo\_prtnr\_ed\_v2\_comb ~   
 demo\_prtnr\_ed\_v2\_comb),  
 highest\_demo\_ed\_bl = case\_when(  
 is.na(demo\_prtnr\_ed\_v2\_bl) == T ~ demo\_prnt\_ed\_v2\_bl,  
 demo\_prnt\_ed\_v2\_bl > demo\_prtnr\_ed\_v2\_bl ~   
 demo\_prnt\_ed\_v2\_bl,  
 demo\_prnt\_ed\_v2\_bl <= demo\_prtnr\_ed\_v2\_bl ~   
 demo\_prtnr\_ed\_v2\_bl))

Create STB variables

abcd\_data <- abcd\_data %>%  
 # aggregate ksad questions to create meaningful, new variables   
 mutate(SI\_ever\_y = case\_when(  
 ksads\_23\_946\_t\_new + ksads\_23\_947\_t\_new + ksads\_23\_948\_t\_new + ksads\_23\_949\_t\_new + ksads\_23\_950\_t\_new > 0 ~ "present",  
 ksads\_23\_957\_t\_new + ksads\_23\_958\_t\_new + ksads\_23\_959\_t\_new + ksads\_23\_960\_t\_new + ksads\_23\_961\_t\_new > 0 ~ "past",  
 is.na(ksads\_23\_946\_t\_new) == T |is.na(ksads\_23\_947\_t\_new) == T |is.na(ksads\_23\_948\_t\_new) == T |   
 is.na(ksads\_23\_949\_t\_new) == T |is.na(ksads\_23\_950\_t\_new) == T |is.na(ksads\_23\_957\_t\_new) == T |  
 is.na(ksads\_23\_958\_t\_new) == T |is.na(ksads\_23\_959\_t\_new) == T |is.na(ksads\_23\_960\_t\_new) == T |  
 is.na(ksads\_23\_961\_t\_new) == T~ NA, # check if vars have NA and preserve the NAs   
 TRUE ~ "never"), # TRUE is case\_when's "else"   
 SI\_ever\_p = case\_when(   
 ksads\_23\_946\_p\_new + ksads\_23\_947\_p\_new + ksads\_23\_948\_p\_new + ksads\_23\_949\_p\_new + ksads\_23\_950\_p\_new > 0 ~ "present",   
 ksads\_23\_957\_p\_new + ksads\_23\_958\_p\_new + ksads\_23\_959\_p\_new + ksads\_23\_960\_p\_new + ksads\_23\_961\_p\_new > 0 ~ "past",  
 is.na(ksads\_23\_946\_p\_new) == T |is.na(ksads\_23\_947\_p\_new) == T |is.na(ksads\_23\_948\_p\_new) == T |  
 is.na(ksads\_23\_949\_p\_new) == T |is.na(ksads\_23\_950\_p\_new) == T |is.na(ksads\_23\_957\_p\_new) == T |  
 is.na(ksads\_23\_958\_p\_new) == T |is.na(ksads\_23\_959\_p\_new) == T |is.na(ksads\_23\_960\_p\_new) == T |  
 is.na(ksads\_23\_961\_p\_new) == T~ NA,  
 TRUE ~ "never"),  
 SI\_ever\_p\_y = case\_when(  
 SI\_ever\_p == "present" | SI\_ever\_y == "present" ~ "present",   
 SI\_ever\_p == "past" | SI\_ever\_y == "past" ~ "past",  
 SI\_ever\_p == "never" | SI\_ever\_y == "never" ~ "never"),  
 SA\_ever\_y = case\_when(  
 is.na(ksads\_23\_954\_t\_new) == T | is.na(ksads\_23\_965\_t\_new) == T ~ NA,  
 ksads\_23\_954\_t\_new > 0 ~ "present",   
 ksads\_23\_965\_t\_new > 0 ~ "past",  
 ksads\_23\_954\_t\_new <= 0 & ksads\_23\_965\_t\_new <= 0 ~ "never"),  
 SA\_ever\_p = case\_when(  
 ksads\_23\_954\_p\_new > 0 ~ "present",  
 ksads\_23\_965\_p\_new > 0 ~ "past",  
 ksads\_23\_954\_p\_new <= 0 & ksads\_23\_965\_p\_new <= 0 ~ "never"),  
 SA\_ever\_p\_y = case\_when(  
 SA\_ever\_p == "present" | SA\_ever\_y == "present" ~ "present",  
 SA\_ever\_p == "past" | SA\_ever\_y == "past" ~ "past",  
 SA\_ever\_p == "never" | SA\_ever\_y == "never" ~ "never"),  
 STB\_highest\_ever\_y = case\_when(  
 is.na(ksads\_23\_954\_t\_new) == T | is.na(ksads\_23\_965\_t\_new) == T |is.na(ksads\_23\_963\_t\_new) == T|   
 is.na(ksads\_23\_964\_t\_new) == T| is.na(ksads\_23\_953\_t\_new)== T | is.na(ksads\_23\_952\_t\_new) == T|  
 is.na(ksads\_23\_949\_t\_new) == T| is.na(ksads\_23\_960\_t\_new) == T|is.na(ksads\_23\_959\_t\_new) == T|  
 is.na(ksads\_23\_961\_t\_new) == T| is.na(ksads\_23\_948\_t\_new) == T| is.na(ksads\_23\_950\_t\_new) == T|  
 is.na(ksads\_23\_947\_t\_new) == T| is.na(ksads\_23\_958\_t\_new) == T| is.na(ksads\_23\_957\_t\_new) == T|  
 is.na(ksads\_23\_946\_t\_new)== T ~ NA,   
   
 0 < (ksads\_23\_954\_t\_new) + (ksads\_23\_965\_t\_new) ~ "SA",  
 0 < (ksads\_23\_963\_t\_new) + (ksads\_23\_964\_t\_new) + (ksads\_23\_953\_t\_new) + (ksads\_23\_952\_t\_new) ~ "SA interrup/aborted",  
 0 < (ksads\_23\_949\_t\_new) + (ksads\_23\_960\_t\_new) ~ "SI intent",  
 0 < (ksads\_23\_959\_t\_new) + (ksads\_23\_961\_t\_new) + (ksads\_23\_948\_t\_new) + (ksads\_23\_950\_t\_new) ~ "SI plan/method",  
 0 < (ksads\_23\_947\_t\_new) + (ksads\_23\_958\_t\_new) ~ "SI active nonspecific",  
 0 < (ksads\_23\_957\_t\_new) + (ksads\_23\_946\_t\_new) ~ "SI passive",  
 TRUE ~ "none"),  
 STB\_highest\_current\_y = case\_when(  
 is.na(ksads\_23\_954\_t\_new) == T | is.na(ksads\_23\_953\_t\_new) == T | is.na(ksads\_23\_952\_t\_new) == T|  
 is.na(ksads\_23\_949\_t\_new) == T| is.na(ksads\_23\_948\_t\_new) == T| is.na(ksads\_23\_950\_t\_new)== T |  
 is.na(ksads\_23\_947\_t\_new)== T | is.na(ksads\_23\_946\_t\_new)== T ~ NA,  
   
 0 < (ksads\_23\_954\_t\_new) ~ "SA",  
 0 < (ksads\_23\_953\_t\_new) + (ksads\_23\_952\_t\_new) ~ "SA interrup/aborted", # in original version of code this level is skipped (all participants in this level became NA)-- in new version it's preserved  
 0 < (ksads\_23\_949\_t\_new) ~ "SI intent",  
 0 < (ksads\_23\_948\_t\_new) + (ksads\_23\_950\_t\_new) ~ "SI plan/method",  
 0 < (ksads\_23\_947\_t\_new) ~ "SI active nonspecific",  
 0 < (ksads\_23\_946\_t\_new) ~ "SI passive",  
 TRUE ~ "none")) %>%  
 mutate(across(c(SI\_ever\_y, SI\_ever\_p,SI\_ever\_p\_y, SA\_ever\_y,SA\_ever\_p,SA\_ever\_p\_y, STB\_highest\_ever\_y,STB\_highest\_current\_y), ~as.factor(.x))) # convert all new vars to factors

Create NSSI Variables

abcd\_data <- abcd\_data %>%  
 mutate(NSSI\_ever\_y = case\_when(  
 is.na(ksads\_23\_945\_t\_new) == T | is.na(ksads\_23\_956\_t\_new) == T ~ NA,  
 ksads\_23\_945\_t\_new > 0 ~ "present",  
 ksads\_23\_956\_t\_new > 0 ~ "past",  
 TRUE ~ "never"),  
 NSSI\_ever\_p = case\_when(  
 is.na(ksads\_23\_945\_p\_new) == T | is.na(ksads\_23\_956\_p\_new) == T ~ NA,  
 ksads\_23\_945\_p\_new > 0 ~ "present",  
 ksads\_23\_956\_p\_new > 0 ~ "past",   
 TRUE ~"never"),  
 NSSI\_ever\_p\_y = case\_when(  
 NSSI\_ever\_p == "present" | NSSI\_ever\_y == "present" ~ "present",  
 NSSI\_ever\_p == "past" | NSSI\_ever\_y == "past" ~ "past",  
 NSSI\_ever\_p == "never" | NSSI\_ever\_y == "never" ~ "never"),   
 SITB\_ever\_y = case\_when(  
 is.na(NSSI\_ever\_y) == T | is.na(STB\_highest\_ever\_y) == T ~ NA,  
 STB\_highest\_ever\_y != "none" | NSSI\_ever\_y != "never" ~ 1,  
 TRUE ~ 0)) %>%  
 mutate(across(c(NSSI\_ever\_y, NSSI\_ever\_p, NSSI\_ever\_p\_y, SITB\_ever\_y), ~as.factor(.x)))

Create Depression Variables

abcd\_data <- abcd\_data %>%  
 mutate(MDD\_ever\_y = case\_when(  
 is.na(ksads\_1\_840\_t\_new) == T |is.na(ksads\_1\_841\_t\_new) == T | is.na(ksads\_1\_842\_t\_new) == T ~ NA,  
 0 < (ksads\_1\_840\_t\_new) ~ "present",  
 0 < (ksads\_1\_841\_t\_new) ~ "partial remission",  
 0< (ksads\_1\_842\_t\_new) ~ "past",  
 TRUE ~ "never"),  
 AnyDD\_ever\_y = case\_when(  
 is.na(ksads\_1\_840\_t\_new) == T |is.na(ksads\_1\_841\_t\_new) == T | is.na(ksads\_1\_842\_t\_new) == T |  
 is.na(ksads\_1\_843\_t\_new) == T | is.na(ksads\_1\_844\_t\_new) == T |is.na(ksads\_1\_845\_t\_new) == T |   
 is.na(ksads\_1\_846\_t\_new) == T | is.na(ksads\_1\_847\_t\_new) == T ~ NA,  
 0 < (ksads\_1\_840\_t\_new) + (ksads\_1\_843\_t\_new) + (ksads\_1\_846\_t\_new) ~ "present",  
 0 < (ksads\_1\_841\_t\_new) + (ksads\_1\_844\_t\_new) ~ "partial remission",  
 0 < (ksads\_1\_842\_t\_new) + (ksads\_1\_845\_t\_new) + (ksads\_1\_847\_t\_new) ~ "past",  
 TRUE ~ "never"),  
 MDD\_ever\_p = case\_when(  
 is.na(ksads\_1\_840\_p\_new) == T |is.na(ksads\_1\_841\_p\_new) == T | is.na(ksads\_1\_842\_p\_new) == T ~ NA,  
 0 < (ksads\_1\_840\_p\_new) ~ "present",  
 0 < (ksads\_1\_841\_p\_new) ~ "partial remission",  
 0 < (ksads\_1\_842\_p\_new) ~ "past",  
 TRUE ~ "never"),  
 AnyDD\_ever\_p = case\_when(  
 is.na(ksads\_1\_840\_p\_new) == T |is.na(ksads\_1\_841\_p\_new) == T | is.na(ksads\_1\_842\_p\_new) == T |  
 is.na(ksads\_1\_843\_p\_new) == T | is.na(ksads\_1\_844\_p\_new) == T |is.na(ksads\_1\_845\_p\_new) == T |   
 is.na(ksads\_1\_846\_p\_new) == T | is.na(ksads\_1\_847\_p\_new) == T ~ NA,  
 0 < (ksads\_1\_840\_p\_new) + (ksads\_1\_843\_p\_new) + (ksads\_1\_846\_p\_new) ~ "present",  
 0 < (ksads\_1\_841\_p\_new) + (ksads\_1\_844\_p\_new) ~ "partial remission",  
 0 < (ksads\_1\_842\_p\_new) + (ksads\_1\_845\_p\_new) + (ksads\_1\_847\_p\_new) ~ "past",  
 TRUE ~ "never"),   
 MDD\_ever\_p\_y = case\_when(  
 MDD\_ever\_p == "present" | MDD\_ever\_y == "present" ~ "present",  
 MDD\_ever\_p == "partial remission" | MDD\_ever\_y == "partial remission" ~ "partial remission",  
 MDD\_ever\_p == "past" | MDD\_ever\_y == "past" ~ "past",  
 MDD\_ever\_p == "never" | MDD\_ever\_y == "never" ~ "never"),  
 AnyDD\_ever\_p\_y = case\_when(  
 AnyDD\_ever\_p == "present" | AnyDD\_ever\_y == "present" ~ "present",  
 AnyDD\_ever\_p == "partial remission" | AnyDD\_ever\_y == "partial remission" ~ "partial remission",  
 AnyDD\_ever\_p == "past" | AnyDD\_ever\_y == "past" ~ "past",  
 AnyDD\_ever\_p == "never" | AnyDD\_ever\_y == "never" ~ "never"  
 )) %>%  
 mutate(across(c(MDD\_ever\_y, AnyDD\_ever\_y, MDD\_ever\_p, AnyDD\_ever\_y, MDD\_ever\_p\_y, AnyDD\_ever\_p\_y), ~as.factor(.x)))

Remove raw variables

abcd\_data <- abcd\_data %>%  
 select(-num\_range("demo\_race\_a\_p\_\_\_", 12:25), # race vars  
 -num\_range("ksads\_23\_", 946:950, "\_t\_new"), # SI, SA, NSSI vars  
 -num\_range("ksads\_23\_", 952:954, "\_t\_new"),  
 -num\_range("ksads\_23\_", 957:961, "\_t\_new"),  
 -num\_range("ksads\_23\_", 963:965, "\_t\_new"),  
 -num\_range("ksads\_23\_", 946:950, "\_p\_new"),  
 -ksads\_23\_954\_p\_new,  
 -num\_range("ksads\_23\_", 957:961, "\_p\_new"),   
 -num\_range("ksads\_1\_", 840:847, "\_t\_new"), # depression vars  
 -num\_range("ksads\_1\_", 840:847, "\_p\_new"))

Remove the time points you do not want

abcd\_data.selected\_time <- abcd\_data %>%  
 filter(eventname %in% timepoint\_list)

Write out a combined data.frame of covariates and outcome variables.

# Creates name based on date and initials/string passed in by user. If no date given, use the current date  
if(is.null(out\_date)){  
 out\_date <- Sys.Date()  
}  
# If no initials/string is given, throw an error that user must input one  
if(is.null(out\_initials)){  
 stop("No input given for 'out\_initials'.   
 User must provide string input to create output file name.")  
}  
# Now output file based on name and out\_dir given. Recall that if out\_dir is NULL, it will write to current wd  
csv\_out\_name <- paste0(out\_dir,'/',out\_date,'\_',out\_initials,'.csv')  
write.csv(abcd\_data.selected\_time, csv\_out\_name, row.names = FALSE)

Here we checkout the columns included in the abcd\_data.selected\_time object. Ideally we would have separate data.frames/ matrices of covariates and outcomes. Furthermore, we observe covariates and outcomes extraneous to our questions, so these need to be removed/ cleaned up to reflect our question.

colnames(abcd\_data.selected\_time)

[1] "src\_subject\_id" "demo\_sex\_v2"   
 [3] "demo\_race\_a\_p\_\_\_10" "demo\_race\_a\_p\_\_\_11"   
 [5] "demo\_race\_a\_p\_\_\_77" "demo\_race\_a\_p\_\_\_99"   
 [7] "demo\_ethn\_v2" "demo\_prnt\_marital\_v2\_bl"   
 [9] "demo\_prnt\_ed\_v2\_bl" "demo\_prtnr\_ed\_v2\_bl"   
 [11] "demo\_comb\_income\_v2\_bl" "rel\_family\_id"   
 [13] "rel\_relationship" "rel\_group\_id"   
 [15] "eventname" "demo\_brthdat\_v2\_comb"   
 [17] "demo\_gender\_id\_v2\_comb" "demo\_prnt\_ed\_v2\_comb"   
 [19] "demo\_prtnr\_ed\_v2\_comb" "demo\_prnt\_marital\_v2\_comb"   
 [21] "demo\_comb\_income\_v2\_comb" "demo\_roster\_v2\_comb"   
 [23] "demo\_fam\_exp1\_v2\_comb" "demo\_fam\_exp2\_v2\_comb"   
 [25] "demo\_fam\_exp3\_v2\_comb" "demo\_fam\_exp4\_v2\_comb"   
 [27] "demo\_fam\_exp5\_v2\_comb" "demo\_fam\_exp6\_v2\_comb"   
 [29] "demo\_fam\_exp7\_v2\_comb" "acs\_raked\_propensity\_score"  
 [31] "kbi\_gender" "kbi\_y\_trans\_id"   
 [33] "kbi\_y\_sex\_orient" "pds\_p\_ss\_male\_category\_2"   
 [35] "pds\_p\_ss\_female\_category\_2" "site\_id\_l"   
 [37] "interview\_age" "mri\_info\_manufacturer"   
 [39] "imgincl\_rsfmri\_include" "prescan\_relaxed\_1"   
 [41] "prescan\_happy\_1" "prescan\_scared\_1"   
 [43] "prescan\_awake\_1" "prescan\_upset\_1"   
 [45] "prescan\_angry\_1" "prescan\_excited\_1"   
 [47] "prescan\_tired\_1" "prescan\_sleepy\_1"   
 [49] "prescan\_sad\_1" "prescan\_relaxed\_2"   
 [51] "prescan\_happy\_2" "prescan\_scared\_2"   
 [53] "prescan\_awake\_2" "prescan\_upset\_2"   
 [55] "prescan\_angry\_2" "prescan\_excited\_2"   
 [57] "prescan\_tired\_2" "prescan\_sleepy\_2"   
 [59] "prescan\_sad\_2" "postscan\_relaxed\_1"   
 [61] "postscan\_happy\_1" "postscan\_afraid\_1"   
 [63] "postscan\_alert\_1" "postscan\_upset\_1"   
 [65] "postscan\_angry\_1" "postscan\_excited\_1"   
 [67] "postscan\_tired\_1" "postscan\_sleepy\_1"   
 [69] "postscan\_sad\_1" "postscan\_relaxed\_2"   
 [71] "postscan\_happy\_2" "postscan\_afraid\_2"   
 [73] "postscan\_alert\_2" "postscan\_upset\_2"   
 [75] "postscan\_angry\_2" "postscan\_excited\_2"   
 [77] "postscan\_tired\_2" "postscan\_sleep\_2"   
 [79] "postscan\_sad\_2" "cbcl\_scr\_syn\_internal\_r"   
 [81] "cbcl\_scr\_syn\_external\_r" "cbcl\_scr\_syn\_totprob\_r"   
 [83] "cbcl\_scr\_dsm5\_depress\_r" "cbcl\_scr\_dsm5\_anxdisord\_r"   
 [85] "cbcl\_scr\_dsm5\_adhd\_r" "cbcl\_scr\_syn\_internal\_t"   
 [87] "cbcl\_scr\_syn\_external\_t" "cbcl\_scr\_syn\_totprob\_t"   
 [89] "cbcl\_scr\_dsm5\_depress\_t" "cbcl\_scr\_dsm5\_anxdisord\_t"   
 [91] "cbcl\_scr\_dsm5\_adhd\_t" "bpm\_y\_scr\_attention\_r"   
 [93] "bpm\_y\_scr\_attention\_t" "bpm\_y\_scr\_internal\_r"   
 [95] "bpm\_y\_scr\_internal\_t" "bpm\_y\_scr\_external\_r"   
 [97] "bpm\_y\_scr\_external\_t" "bpm\_y\_scr\_totalprob\_r"   
 [99] "bpm\_y\_scr\_totalprob\_t" "ksads\_23\_945\_t\_new"   
[101] "ksads\_23\_956\_t\_new" "ksads\_23\_144\_t\_new"   
[103] "ksads\_23\_143\_t\_new" "ksads\_23\_962\_t\_new"   
[105] "ksads\_23\_951\_t\_new" "ksads\_23\_955\_t\_new"   
[107] "ksads\_23\_966\_t\_new" "ksads\_23\_945\_p\_new"   
[109] "ksads\_23\_956\_p\_new" "ksads\_23\_144\_p\_new"   
[111] "ksads\_23\_143\_p\_new" "ksads\_23\_955\_p\_new"   
[113] "ksads\_23\_966\_p\_new" "ksads\_23\_962\_p\_new"   
[115] "ksads\_23\_965\_p\_new" "ksads\_23\_951\_p\_new"   
[117] "demo\_brthdat\_v2\_comb\_clean" "interview\_age\_b"   
[119] "White\_race" "Black\_race"   
[121] "AIAN\_race" "NHPI\_race"   
[123] "Asian\_race" "Other\_race"   
[125] "Missing\_race" "Indigenous\_race"   
[127] "pubertal\_status" "highest\_demo\_ed\_comb"   
[129] "highest\_demo\_ed\_bl" "SI\_ever\_y"   
[131] "SI\_ever\_p" "SI\_ever\_p\_y"   
[133] "SA\_ever\_y" "SA\_ever\_p"   
[135] "SA\_ever\_p\_y" "STB\_highest\_ever\_y"   
[137] "STB\_highest\_current\_y" "NSSI\_ever\_y"   
[139] "NSSI\_ever\_p" "NSSI\_ever\_p\_y"   
[141] "SITB\_ever\_y" "MDD\_ever\_y"   
[143] "AnyDD\_ever\_y" "MDD\_ever\_p"   
[145] "AnyDD\_ever\_p" "MDD\_ever\_p\_y"   
[147] "AnyDD\_ever\_p\_y"

### Neuroimaging

#### sMRI: Cortical Thickness (CT) and Surface Area (SA)

#### fMRI: Functional Connectivity (FC)

### Genomics/ epigenomics

## Simulation Topic

## Relevant References

## Appendix: All code for this document consolidated

library(tidyverse)  
library(psych) # For polychoric correlation  
library(Rtsne) # For tsne plots  
library(VennDiagram) # For venn diagram comparing variables used in 2 diff ELA papers  
# Set the path for raw data files  
data\_dir <- '/Users/aidanneher/Library/CloudStorage/Box-Box/ABCD Tabulated Data/5.1/core'  
# Location of desired output directory - if NULL, will output into working directory  
out\_dir <- '/Users/aidanneher/Documents/GitHub/abcd\_multiview/data'  
# out\_dir <- NULL  
# Date you used in output name - if NULL, will use output from Sys.Date() (current date)  
# e.g. out\_date <- '2023-03-03'   
out\_date <- NULL  
# Initials or other string you want in output naming - no NULL option here  
out\_initials <- 'AN'   
# Defaults to baseline data - alternatively, pass a vector of eventnames  
load\_ela\_data <- function(data\_dir, events = "baseline\_year\_1\_arm\_1") {  
   
 # Physical and sexual violence  
 mh\_p\_ksads\_path <- file.path(data\_dir, "mental-health/mh\_p\_ksads\_ptsd.csv")  
 mh\_p\_ksads <- read\_csv(mh\_p\_ksads\_path) %>%  
 select(src\_subject\_id, eventname, ksads\_ptsd\_raw\_761\_p,  
 ksads\_ptsd\_raw\_762\_p, ksads\_ptsd\_raw\_763\_p,  
 ksads\_ptsd\_raw\_767\_p, ksads\_ptsd\_raw\_768\_p,  
 ksads\_ptsd\_raw\_769\_p, ksads\_ptsd\_raw\_760\_p,  
 ksads\_ptsd\_raw\_764\_p, ksads\_ptsd\_raw\_765\_p)  
   
 # Parent psychopathology  
 mh\_p\_fhx\_path <- file.path(data\_dir, "mental-health/mh\_p\_fhx.csv")  
 mh\_p\_fhx <- read\_csv(mh\_p\_fhx\_path) %>%  
 select(src\_subject\_id, eventname, famhx\_4\_p,  
 fam\_history\_5\_yes\_no, fam\_history\_6\_yes\_no,  
 fam\_history\_7\_yes\_no, fam\_history\_8\_yes\_no,  
 fam\_history\_11\_yes\_no, fam\_history\_12\_yes\_no,  
 fam\_history\_13\_yes\_no)  
   
 # Neighborhood Threat  
 ce\_p\_nsc\_path <- file.path(data\_dir, "culture-environment/ce\_p\_nsc.csv")  
 ce\_p\_nsc <- read\_csv(ce\_p\_nsc\_path) %>%  
 select(src\_subject\_id, eventname, neighborhood3r\_p, neighborhood2r\_p)  
   
 # Prenatal Substance Exposure  
 ph\_p\_dhx\_path <- file.path(data\_dir, "physical-health/ph\_p\_dhx.csv")  
 ph\_p\_dhx <- read\_csv(ph\_p\_dhx\_path) %>%  
 select(src\_subject\_id, eventname,  
 devhx\_9\_tobacco, devhx\_9\_alcohol,  
 devhx\_9\_marijuana, devhx\_9\_coc\_crack,  
 devhx\_9\_her\_morph, devhx\_9\_oxycont)  
   
 # Scarcity  
 abcd\_p\_demo\_path <- file.path(data\_dir, "abcd-general/abcd\_p\_demo.csv")  
 abcd\_p\_demo <- read\_csv(abcd\_p\_demo\_path) %>%  
 select(src\_subject\_id, eventname, demo\_fam\_exp1\_v2, demo\_fam\_exp5\_v2)  
   
 # Household Dysfunction  
 ce\_y\_fes\_path <- file.path(data\_dir, "culture-environment/ce\_y\_fes.csv")  
 ce\_y\_fes <- read\_csv(ce\_y\_fes\_path) %>%  
 select(src\_subject\_id, eventname, fes\_youth\_q6, fes\_youth\_q1, fes\_youth\_q5)  
   
 # Merging all data frames  
 merged\_data <- list(mh\_p\_ksads, mh\_p\_fhx, ce\_p\_nsc, ph\_p\_dhx, abcd\_p\_demo, ce\_y\_fes) %>%  
 reduce(full\_join, by = c("src\_subject\_id", "eventname"))  
   
 # Filter to events of interest  
 filtered\_data <- merged\_data %>%  
 filter(eventname %in% events)  
   
 # Recode values in all columns except 'src\_subject\_id' and 'eventname'  
 tidy\_data <- filtered\_data %>%  
 mutate(across(-c(src\_subject\_id, eventname), ~na\_if(., 999))) %>% # Recode 999 "Don't know" as NA  
 mutate(across(-c(src\_subject\_id, eventname), ~na\_if(., 7))) %>% # Recode 7 "Refuse to answer" as NA  
 mutate(across(-c(src\_subject\_id, eventname), ~na\_if(., 777))) # Recode 777 "Refuse to answer" as NA  
   
 return(tidy\_data)  
}  
  
ela\_data <- load\_ela\_data(data\_dir)  
  
  
str(ela\_data)  
  
  
# Calculate the proportion of missing values per column, excluding specific columns  
missing\_data <- ela\_data %>%  
 select(-src\_subject\_id, -eventname) %>% # Exclude these columns from the analysis  
 summarise(across(everything(), ~sum(is.na(.))/n())) %>%  
 pivot\_longer(cols = everything(), names\_to = "variable", values\_to = "prop\_missing")  
  
# Plot the proportion of missing data  
ggplot(missing\_data, aes(x = variable, y = prop\_missing)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 labs(x = "Variable", y = "Proportion Missing", title = "Proportion of Missing Data by ELA Variable at Baseline")  
  
  
# Calculate the proportion of non-zero values per column, excluding specific columns  
non\_zero\_data <- ela\_data %>%  
 select(-src\_subject\_id, -eventname) %>% # Exclude these columns from the analysis  
 summarise(across(everything(), ~mean(. != 0, na.rm = TRUE))) %>%  
 pivot\_longer(cols = everything(), names\_to = "variable", values\_to = "prop\_non\_zero")  
  
# Plot the proportion of non-zero data  
ggplot(non\_zero\_data, aes(x = variable, y = prop\_non\_zero)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 geom\_hline(yintercept = 0.05, linetype = "dashed", color = "red") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 labs(x = "Variable", y = "Proportion Non-Zero",   
 title = "Proportion of Non-Zero Observations by ELA Variable at Baseline")  
  
print(paste("N Variables with zero variance:", sum(non\_zero\_data$prop\_non\_zero == 0)))  
  
# Prepare the data by excluding non-relevant columns and ensuring complete cases  
filtered\_data <- ela\_data %>%  
 select(-src\_subject\_id, -eventname) %>%  
 na.omit()  
  
# Filter out variables with more than 2 unique levels  
binary\_data <- filtered\_data %>%  
 select\_if(~length(unique(.)) == 2)  
  
# Compute the polychoric correlation, only for binary variables  
polychoric\_matrix <- polychoric(binary\_data)$rho  
  
# Transform the correlation matrix for visualization  
correlation\_data <- as.data.frame(polychoric\_matrix) %>%  
 rownames\_to\_column("Variable1") %>%  
 pivot\_longer(cols = -Variable1, names\_to = "Variable2", values\_to = "Correlation")  
  
# Plot the correlation matrix as a heatmap  
ggplot(correlation\_data, aes(x = Variable1, y = Variable2, fill = Correlation)) +  
 geom\_tile() +  
 scale\_fill\_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1)) +  
 labs(fill = "Correlation", title = "Polychoric Correlation Matrix of Binary ELA Variables", x = NULL, y = NULL)  
  
# Organize variable pairs to ensure unique combinations  
high\_correlation\_pairs <- correlation\_data %>%  
 mutate(VariablePair = pmap\_chr(list(Variable1, Variable2), ~paste(sort(c(...)), collapse = "-"))) %>%  
 distinct(VariablePair, .keep\_all = TRUE) %>%  
 filter(abs(Correlation) > 0.75, Variable1 != Variable2)  
  
# View the table of feature pairs "VariablePair" with high correlations  
high\_correlation\_pairs %>%  
 select(VariablePair, Correlation) %>%  
 print  
# Remove duplicate rows  
unique\_binary\_data <- binary\_data %>%   
 distinct()  
  
# Check the number of rows removed  
cat("Removed", nrow(binary\_data) - nrow(unique\_binary\_data), "duplicate rows.\n")  
  
# Set parameters for t-SNE  
set.seed(42) # for reproducibility  
tsne\_results <- Rtsne(as.matrix(unique\_binary\_data), dims = 2, perplexity = 30, verbose = TRUE, max\_iter = 500)  
  
# Create a data frame for plotting  
tsne\_data <- as.data.frame(tsne\_results$Y)  
colnames(tsne\_data) <- c("TSNE1", "TSNE2")  
  
# Plot t-SNE results  
ggplot(tsne\_data, aes(x = TSNE1, y = TSNE2)) +  
 geom\_point(alpha = 0.5) +  
 theme\_minimal() +  
 labs(title = "t-SNE Visualization of ELA Binary Variables",  
 x = "t-SNE Dimension 1",  
 y = "t-SNE Dimension 2")  
# Find duplicates (except for the first occurrence)  
duplicate\_data <- binary\_data %>%  
 add\_count() %>%   
 filter(n > 1) %>%  
 select(-n) %>%  
 distinct()  
  
# Count how many times each unique set of responses appears  
duplicate\_summary <- binary\_data %>%  
 group\_by(across(everything())) %>%  
 summarise(count = n(), .groups = 'drop') %>%  
 arrange(desc(count))  
  
# # Look at the most common duplicate entries  
# print(duplicate\_summary)  
  
# # For visualization, you may want to check specific questions or all questions  
# # Here's a general approach for visualizing the distribution of responses for one variable:  
# ggplot(duplicate\_summary, aes(x = ksads\_ptsd\_raw\_761\_p, y = count)) +  
# geom\_col() +  
# labs(title = "Distribution of Responses for a Specific Variable in Duplicate Rows",  
# x = "Response",  
# y = "Count of Duplicates")  
  
# Or visualize all variables' responses in duplicates  
duplicate\_summary\_long <- pivot\_longer(duplicate\_summary, cols = -count,   
 names\_to = "variable", values\_to = "response")  
  
ggplot(duplicate\_summary\_long, aes(x = response, fill = as.factor(response))) +  
 geom\_histogram(stat = "count") +  
 facet\_wrap(~ variable, scales = "free\_x") +  
 theme\_minimal() +  
 labs(title = "Distribution of Responses Across All Variables in Duplicate Rows",  
 x = "Response",  
 y = "Count")  
# Read in family variables  
rel <- read.csv(paste(data\_dir, 'abcd-general/abcd\_y\_lt.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, rel\_family\_id)  
  
# Read in family substance use summary scores  
fhx <- read.csv(paste(data\_dir, 'mental-health/mh\_p\_fhx.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, famhx\_ss\_fath\_prob\_alc\_p, famhx\_ss\_moth\_prob\_alc\_p, famhx\_ss\_fath\_prob\_dg\_p, famhx\_ss\_moth\_prob\_dg\_p)  
  
# Read in parent demographics  
pdemo <- read.csv(paste(data\_dir, 'abcd-general/abcd\_p\_demo.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, demo\_prim, demo\_prnt\_marital\_v2, demo\_prnt\_ed\_v2, demo\_prtnr\_ed\_v2, demo\_comb\_income\_v2,  
 demo\_fam\_exp1\_v2, demo\_fam\_exp2\_v2, demo\_fam\_exp3\_v2, demo\_fam\_exp4\_v2, demo\_fam\_exp5\_v2,   
 demo\_fam\_exp6\_v2, demo\_fam\_exp7\_v2)  
  
# Read in CRPBI  
crpbi <- read.csv(paste(data\_dir, 'culture-environment/ce\_y\_crpbi.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, crpbi\_parent1\_y, crpbi\_caregiver12\_y, crpbi\_parent2\_y, crpbi\_caregiver13\_y,  
 crpbi\_parent3\_y, crpbi\_caregiver14\_y, crpbi\_parent4\_y, crpbi\_caregiver15\_y, crpbi\_parent5\_y,   
 crpbi\_caregiver16\_y)  
  
# Read in parent report family environment scale  
fes02 <- read.csv(paste(data\_dir, 'culture-environment/ce\_p\_fes.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, fam\_enviro1\_p, fam\_enviro2r\_p, fam\_enviro3\_p, fam\_enviro4r\_p, fam\_enviro5\_p,  
 fam\_enviro6\_p, fam\_enviro7r\_p, fam\_enviro8\_p, fam\_enviro9r\_p)  
  
# Read in youth report family environment scale  
fes01 <- read.csv(paste(data\_dir, 'culture-environment/ce\_y\_fes.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, fes\_youth\_q1, fes\_youth\_q2, fes\_youth\_q3, fes\_youth\_q4, fes\_youth\_q5, fes\_youth\_q6,  
 fes\_youth\_q7, fes\_youth\_q8, fes\_youth\_q9)  
  
# Read in ksads trauma, parent interview  
ptsd <- read.csv(paste(data\_dir, 'mental-health/mh\_p\_ksads\_ptsd.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, ksads\_ptsd\_raw\_754\_p, ksads\_ptsd\_raw\_755\_p, ksads\_ptsd\_raw\_756\_p, ksads\_ptsd\_raw\_757\_p,  
 ksads\_ptsd\_raw\_758\_p, ksads\_ptsd\_raw\_759\_p, ksads\_ptsd\_raw\_760\_p, ksads\_ptsd\_raw\_761\_p,  
 ksads\_ptsd\_raw\_762\_p, ksads\_ptsd\_raw\_763\_p, ksads\_ptsd\_raw\_764\_p, ksads\_ptsd\_raw\_765\_p,  
 ksads\_ptsd\_raw\_766\_p, ksads\_ptsd\_raw\_767\_p, ksads\_ptsd\_raw\_768\_p, ksads\_ptsd\_raw\_769\_p,  
 ksads\_ptsd\_raw\_770\_p)  
  
# Read in parental monitoring  
pmq <- read.csv(paste(data\_dir, 'culture-environment/ce\_y\_pm.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, parent\_monitor\_q1\_y, parent\_monitor\_q2\_y, parent\_monitor\_q3\_y, parent\_monitor\_q4\_y,  
 parent\_monitor\_q5\_y)  
  
# Read in neighborhood safety and crime, parents  
pnscss <- read.csv(paste(data\_dir, 'culture-environment/ce\_p\_nsc.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, nsc\_p\_ss\_mean\_3\_items)  
  
# Read in neighborhood safety and crime, youth  
ynsc <- read.csv(paste(data\_dir, 'culture-environment/ce\_y\_nsc.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, neighborhood\_crime\_y)  
  
# Read in ASR (parent psychopathology)  
asr <- read.csv(paste(data\_dir, 'mental-health/mh\_p\_asr.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, asr\_scr\_anxdisord\_r, asr\_scr\_somaticpr\_r, asr\_scr\_depress\_r, asr\_scr\_avoidant\_r,  
 asr\_scr\_adhd\_r, asr\_scr\_antisocial\_r, asr\_scr\_inattention\_r, asr\_scr\_hyperactive\_r)  
  
# Read in ADI data file  
ADI <- read.csv(paste(data\_dir, 'linked-external-data/led\_l\_adi.csv', sep='/')) %>%  
 filter(eventname == "baseline\_year\_1\_arm\_1") %>%   
 select(src\_subject\_id, reshist\_addr1\_adi\_wsum)  
  
# Merge all data frames  
ela\_data\_brieant <- full\_join(fhx, asr) %>%  
 full\_join(pdemo) %>%  
 full\_join(crpbi) %>%  
 full\_join(fes01) %>%  
 full\_join(fes02) %>%  
 full\_join(ptsd) %>%  
 full\_join(pmq) %>%  
 full\_join(pnscss) %>%  
 full\_join(ynsc) %>%  
 full\_join(rel) %>%  
 full\_join(ADI)  
  
print(paste("ncols of ELA data used for Brieant el al. 2023 analysis:",  
 ncol(ela\_data\_brieant)))  
# Exclude specified variables and extract variable names from each dataset  
vars\_ela\_orendain <- setdiff(names(ela\_data),   
 c("src\_subject\_id", "eventname", "rel\_family\_id"))  
vars\_ela\_brieant <- setdiff(names(ela\_data\_brieant),   
 c("src\_subject\_id", "eventname", "rel\_family\_id"))  
  
# Draw a Venn diagram  
venn.plot <- venn.diagram(  
 x = list(  
 Orendain = vars\_ela\_orendain,  
 Brieant = vars\_ela\_brieant  
 ),  
 filename = NULL, # Set to NULL for plotting directly in R  
 category.names = c("Orendain et al. 2023", "Brieant et al. 2023"),  
 output = TRUE,  
 main = "Venn Diagram of Variables in ELA Data Studies",  
 fill = c("salmon", "lightblue"),  
 alpha = 0.50,  
 label.col = "darkblue",  
 cex = 1.5,  
 fontfamily = "serif",  
 cat.cex = 1.5,  
 cat.col = "darkblue",  
 cat.pos = 0,  
 cat.dist = 0.07,  
 cat.fontfamily = "serif"  
)  
  
# Display the Venn diagram  
grid.draw(venn.plot)  
  
# For Ellery's code, we must specify what time points we want.   
# Put the string provided for all desired timepoints into a vector e.g. c('timepoint1\_name','timepoint2\_name') etc.  
timepoint\_list <- c("baseline\_year\_1\_arm\_1")   
# Function to load the required datasets  
load\_datasets <- function(data\_dir) {  
 # Standard files for most studies  
 demog <- read\_csv(file.path(data\_dir, "abcd-general/abcd\_p\_demo.csv"))  
 demog\_y <- read\_csv(file.path(data\_dir, "mental-health/mh\_y\_ksads\_bg.csv"))  
 puberty <- read\_csv(file.path(data\_dir, "physical-health/ph\_p\_pds.csv"))  
 study\_covars <- read\_csv(file.path(data\_dir, "abcd-general/abcd\_y\_lt.csv"))  
 sib\_twin <- read\_csv(file.path(data\_dir, "genetics/gen\_y\_pihat.csv"))  
   
 # MRI standard files  
 mri <- read\_csv(file.path(data\_dir, "imaging/mri\_y\_adm\_info.csv"))  
 qc <- read\_csv(file.path(data\_dir, "imaging/mri\_y\_qc\_incl.csv"))  
 scan\_qtns <- read\_csv(file.path(data\_dir, "imaging/mri\_y\_adm\_qtn.csv"))  
   
 # Standard files for clinical data  
 cbcl <- read\_csv(file.path(data\_dir, "mental-health/mh\_p\_cbcl.csv"))  
 bpm\_y <- read\_csv(file.path(data\_dir, "mental-health/mh\_y\_bpm.csv"))  
 ksad\_p <- read\_csv(file.path(data\_dir, "mental-health/mh\_p\_ksads\_ss.csv"))  
 ksad\_y <- read\_csv(file.path(data\_dir, "mental-health/mh\_y\_ksads\_ss.csv"))  
   
 # Combine all datasets into a list (if needed for further processing or return)  
 datasets <- list(demog = demog, demog\_y = demog\_y, puberty = puberty,  
 study\_covars = study\_covars, sib\_twin = sib\_twin, mri = mri,  
 qc = qc, scan\_qtns = scan\_qtns, cbcl = cbcl, bpm\_y = bpm\_y,  
 ksad\_p = ksad\_p, ksad\_y = ksad\_y)  
   
 return(datasets)  
}  
  
# Load the datasets  
datasets <- load\_datasets(data\_dir)  
  
# Make data.frames accessible to global environment   
attach(datasets)  
  
# KSAD INITIAL CLEANING  
# Why? During the data collection, a new version of the KSAD was used (KSAD2), so each KSAD variable has a corresponding KSAD2 variable which contains the data since the new version was adopted. The code below combines these variables into one.  
  
ksad\_new\_df <- ksad\_p %>% # initialize new data frame  
 select(src\_subject\_id, eventname)  
   
clean\_ksads <- function(ksad\_new\_df, ksad, ksad2, p\_or\_t){   
 # goal: combine two ksad variables into one new var  
 # ksad\_new\_df = a new data frame created above to hold the new vars,   
 # ksad = 1st ksad variable (old ksad),   
 # ksad2 = ksad2 var (same content/question as ksad1),   
 # p\_or\_t = whether the variable is a p or t ksad   
 # output: a dataframe containing the new variable  
 assign("ksad", paste0(ksad, sep = "\_", p\_or\_t)) # create ksad variable from inputs and assign it the name ksad  
 assign("ksad2", paste0(ksad2, sep = "\_", p\_or\_t)) # create ksad2 variable from inputs and assign it the name ksad2  
 assign("ksad\_new", paste0(ksad, "\_new")) # create new variable name from inputs and assign it the name ksad\_new  
 assign("ksad\_df", paste("ksad", if\_else(p\_or\_t == "p", "p", "y"), sep = "\_")) # create data frame name from inputs and name it ksad\_df  
 my\_cols <- c("src\_subject\_id", "eventname", ksad, ksad2) # create vector of necessary column names  
 intermediary <- get(ksad\_df) %>%  
 select(all\_of(my\_cols)) %>% # select cols (need to do it this way because I'm referring to the cols as strings)  
 mutate(!!ksad\_new := if\_else(is.na(get(ksad)) == T, get(ksad2), get(ksad))) # create new var  
 ksad\_new\_df <- right\_join(intermediary, ksad\_new\_df, by = c("src\_subject\_id", "eventname")) # join new var to existing data set  
 ksad\_new\_df <- ksad\_new\_df %>%  
 select(src\_subject\_id, eventname, ends\_with("\_new")) # select only necessary vars  
 return(ksad\_new\_df)  
}  
  
  
# call the function with each pair of ksad and ksad2 variables (make sure to assign the output to the same name so you create one data set with all the new variables)  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_946", "ksads2\_23\_906", "p") # suicidal ideation  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_947", "ksads2\_23\_907", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_948", "ksads2\_23\_908", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_949", "ksads2\_23\_909", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_950", "ksads2\_23\_910", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_951", "ksads2\_23\_911", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_957", "ksads2\_23\_917", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_958", "ksads2\_23\_918", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_959", "ksads2\_23\_919", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_960", "ksads2\_23\_920", "p")   
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_961", "ksads2\_23\_921", "p")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_954", "ksads2\_23\_914", "p") # suicidal attempts, prep  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_965", "ksads2\_23\_925", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_962", "ksads2\_23\_922", "p")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_966", "ksads2\_23\_926", "p") # NO suicidal ideation/behaviors  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_955", "ksads2\_23\_915", "p")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_143", "ksads2\_23\_134", "p") # non-suicidal self injury  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_144", "ksads2\_23\_135", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_956", "ksads2\_23\_916", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_945", "ksads2\_23\_905", "p")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_840", "ksads2\_1\_790", "p") # depression  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_841", "ksads2\_1\_791", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_842", "ksads2\_1\_792", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_843", "ksads2\_1\_793", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_844", "ksads2\_1\_794", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_845", "ksads2\_1\_795", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_846", "ksads2\_1\_796", "p")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_847", "ksads\_1\_847", "p") # this one does not have a corresponding ksad2 (I put it here so it would still appear in the df)  
  
# repeat function calls with t instead of p  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_946", "ksads2\_23\_906", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_947", "ksads2\_23\_907", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_948", "ksads2\_23\_908", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_949", "ksads2\_23\_909", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_950", "ksads2\_23\_910", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_957", "ksads2\_23\_917", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_958", "ksads2\_23\_918", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_959", "ksads2\_23\_919", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_960", "ksads2\_23\_920", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_961", "ksads2\_23\_921", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_954", "ksads2\_23\_914", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_965", "ksads2\_23\_925", "t")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_966", "ksads2\_23\_926", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_955", "ksads2\_23\_915", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_951", "ksads2\_23\_911", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_962", "ksads2\_23\_922", "t")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_143", "ksads2\_23\_134", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_144", "ksads2\_23\_135", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_956", "ksads2\_23\_916", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_945", "ksads2\_23\_905", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_963", "ksads2\_23\_923", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_964", "ksads2\_23\_924", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_953", "ksads2\_23\_913", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_23\_952", "ksads2\_23\_912", "t")  
  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_840", "ksads2\_1\_790", "t")   
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_841", "ksads2\_1\_791", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_842", "ksads2\_1\_792", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_843", "ksads2\_1\_793", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_844", "ksads2\_1\_794", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_845", "ksads2\_1\_795", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_846", "ksads2\_1\_796", "t")  
ksad\_new\_df <- clean\_ksads(ksad\_new\_df, "ksads\_1\_847", "ksads\_1\_847", "t") # this one does not have a corresponding ksad2 (I put it here so it would still appear in the df)  
# COMBINE BASELINE VARIABLES WITH LONGITUDINAL VARIABLES  
# how it combines: if the observation occurred in year1/arm1 (the baseline) the combination variable takes the value of the baseline variable, if not the combo variable takes the value of the longitudinal variable  
var\_vect <- c("demo\_brthdat\_v2", "demo\_gender\_id\_v2", "demo\_prnt\_ed\_v2", "demo\_prtnr\_ed\_v2" , "demo\_prnt\_marital\_v2", "demo\_comb\_income\_v2", "demo\_roster\_v2", "demo\_fam\_exp1\_v2", "demo\_fam\_exp2\_v2", "demo\_fam\_exp3\_v2", "demo\_fam\_exp4\_v2", "demo\_fam\_exp5\_v2", "demo\_fam\_exp6\_v2", "demo\_fam\_exp7\_v2") # input any baseline variable name you wish to combine with long. var in this vector  
combine.vars <- function(var){ # takes baseline variable name (a string) and makes a new variable using the corresponding long. var, outputs demog dataset with new combo variables (combo variables will be named the baseline variable name with "\_comb" at the end)  
 assign("var\_comb", paste0(var, "\_comb"))  
 assign("var\_l", paste0(var, "\_l"))  
 demog <- demog %>%  
 mutate(!!var\_comb := if\_else(eventname == "baseline\_year\_1\_arm\_1", get(var), get(var\_l))) # if observation occurred in year 1, arm 1 use bl var if not use long var  
 return(demog)  
}  
  
for (i in 1:length(var\_vect)) { # loops through vector of all variables to be combined and applies the combine.vars function  
 demog <- combine.vars(var\_vect[i])}  
demog\_bl <- demog[demog$eventname == "baseline\_year\_1\_arm\_1",]  
  
demog\_bl <- demog\_bl %>%   
 select(src\_subject\_id, demo\_sex\_v2, demo\_race\_a\_p\_\_\_10,  
 demo\_race\_a\_p\_\_\_11,demo\_race\_a\_p\_\_\_12, demo\_race\_a\_p\_\_\_13, demo\_race\_a\_p\_\_\_14,  
 demo\_race\_a\_p\_\_\_15, demo\_race\_a\_p\_\_\_16, demo\_race\_a\_p\_\_\_17, demo\_race\_a\_p\_\_\_18,  
 demo\_race\_a\_p\_\_\_19, demo\_race\_a\_p\_\_\_20, demo\_race\_a\_p\_\_\_21, demo\_race\_a\_p\_\_\_22,  
 demo\_race\_a\_p\_\_\_23, demo\_race\_a\_p\_\_\_24, demo\_race\_a\_p\_\_\_25,demo\_race\_a\_p\_\_\_77,   
 demo\_race\_a\_p\_\_\_99, demo\_ethn\_v2, demo\_prnt\_marital\_v2, demo\_prnt\_ed\_v2, demo\_prtnr\_ed\_v2, demo\_comb\_income\_v2)  
  
demog\_bl <- dplyr::rename(demog\_bl, demo\_prnt\_marital\_v2\_bl = demo\_prnt\_marital\_v2)  
demog\_bl <- dplyr::rename(demog\_bl, demo\_comb\_income\_v2\_bl = demo\_comb\_income\_v2)  
demog\_bl <- dplyr::rename(demog\_bl, demo\_prnt\_ed\_v2\_bl = demo\_prnt\_ed\_v2)  
demog\_bl <- dplyr::rename(demog\_bl, demo\_prtnr\_ed\_v2\_bl = demo\_prtnr\_ed\_v2)  
  
demog <- demog %>%  
 select(src\_subject\_id, eventname, demo\_brthdat\_v2\_comb, demo\_gender\_id\_v2\_comb,   
 demo\_prnt\_ed\_v2\_comb, demo\_prtnr\_ed\_v2\_comb, demo\_prnt\_marital\_v2\_comb, demo\_comb\_income\_v2\_comb,  
 demo\_roster\_v2\_comb, demo\_fam\_exp1\_v2\_comb, demo\_fam\_exp2\_v2\_comb, demo\_fam\_exp3\_v2\_comb,   
 demo\_fam\_exp4\_v2\_comb, demo\_fam\_exp5\_v2\_comb, demo\_fam\_exp6\_v2\_comb, demo\_fam\_exp7\_v2\_comb,  
 acs\_raked\_propensity\_score)  
  
demog\_y <- demog\_y %>%  
 select(src\_subject\_id, eventname, kbi\_gender, kbi\_y\_trans\_id, kbi\_y\_sex\_orient)  
  
puberty <- puberty %>%  
 select(src\_subject\_id, eventname, pds\_p\_ss\_male\_category\_2, pds\_p\_ss\_female\_category\_2)  
  
fam <- study\_covars[study\_covars$eventname == "baseline\_year\_1\_arm\_1",]  
  
fam <- fam %>%  
 select(src\_subject\_id, rel\_family\_id)  
  
study\_covars <- study\_covars %>%  
 select(src\_subject\_id, eventname, site\_id\_l, interview\_age)  
  
sib\_twin <- sib\_twin %>%   
 select(src\_subject\_id, rel\_relationship, rel\_group\_id)  
  
mri <- mri %>%  
 select(src\_subject\_id, eventname, mri\_info\_manufacturer)  
  
qc <- qc %>%  
 select(src\_subject\_id, eventname, imgincl\_rsfmri\_include)  
  
cbcl <- cbcl %>%  
 select(src\_subject\_id, eventname, cbcl\_scr\_syn\_internal\_r, cbcl\_scr\_syn\_external\_r, cbcl\_scr\_syn\_totprob\_r,  
 cbcl\_scr\_dsm5\_depress\_r, cbcl\_scr\_dsm5\_anxdisord\_r, cbcl\_scr\_dsm5\_adhd\_r,  
 cbcl\_scr\_syn\_internal\_t, cbcl\_scr\_syn\_external\_t, cbcl\_scr\_syn\_totprob\_t,  
 cbcl\_scr\_dsm5\_depress\_t, cbcl\_scr\_dsm5\_anxdisord\_t, cbcl\_scr\_dsm5\_adhd\_t)  
  
bpm\_y <- bpm\_y %>%   
 select(src\_subject\_id, eventname, bpm\_y\_scr\_attention\_r, bpm\_y\_scr\_attention\_t, bpm\_y\_scr\_internal\_r,   
 bpm\_y\_scr\_internal\_t, bpm\_y\_scr\_external\_r, bpm\_y\_scr\_external\_t, bpm\_y\_scr\_totalprob\_r,   
 bpm\_y\_scr\_totalprob\_t)  
  
  
files <- list(demog, demog\_y, puberty, study\_covars,   
 mri, qc, scan\_qtns, cbcl, bpm\_y, ksad\_new\_df)  
abcd\_data\_0 <- files %>% reduce(full\_join, by = c("src\_subject\_id", "eventname"))  
  
files\_2 <- list(demog\_bl, fam, sib\_twin, abcd\_data\_0)  
abcd\_data <- files\_2 %>% reduce(full\_join, by = "src\_subject\_id")  
  
abcd\_data <- abcd\_data %>%  
 mutate(across(where(is.numeric), ~na\_if(.,777))) %>%  
 mutate(across(where(is.numeric), ~na\_if(.,999))) %>%  
 mutate(across(where(is.numeric), ~na\_if(.,555))) %>%  
 mutate(across(where(is.numeric), ~na\_if(.,888))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"777"))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"999"))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"555"))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"888"))) %>%  
 mutate(across(where(is.character), ~na\_if(.,"")))   
  
abcd\_data <- abcd\_data %>%  
 mutate(demo\_brthdat\_v2\_comb\_clean = if\_else(demo\_brthdat\_v2\_comb > 21,   
 demo\_brthdat\_v2\_comb/12, demo\_brthdat\_v2\_comb), # convert months to years  
 demo\_brthdat\_v2\_comb\_clean = if\_else(demo\_brthdat\_v2\_comb\_clean < 8, NA, demo\_brthdat\_v2\_comb\_clean), # younger than 8 --> NA  
 demo\_brthdat\_v2\_comb\_clean = trunc(demo\_brthdat\_v2\_comb\_clean)) %>% # remove decimals   
 mutate(interview\_age\_b = interview\_age / 12) # convert months to years  
abcd\_data <- abcd\_data %>%  
 mutate(demo\_ethn\_v2 = abs(demo\_ethn\_v2 - 2), # change "2" to 0 to match other vars   
 White\_race = demo\_race\_a\_p\_\_\_10,  
 Black\_race = demo\_race\_a\_p\_\_\_11,  
 AIAN\_race = if\_else(rowSums(select(., num\_range("demo\_race\_a\_p\_\_\_", 12:13)))  
 >=1, 1, 0),  
 NHPI\_race = if\_else(rowSums(select(., num\_range("demo\_race\_a\_p\_\_\_", 14:17)))  
 >=1, 1, 0),  
 Asian\_race = if\_else(rowSums(select(., num\_range("demo\_race\_a\_p\_\_\_", 18:24)))  
 >=1, 1, 0),  
 Other\_race = demo\_race\_a\_p\_\_\_25,  
 Missing\_race = if\_else(demo\_race\_a\_p\_\_\_99 == 1, 1, demo\_race\_a\_p\_\_\_77),   
 Missing\_race = if\_else(White\_race == 1 | # if participant did not endorse any of these, mark as missing  
 Black\_race == 1 |   
 AIAN\_race == 1 |  
 NHPI\_race == 1 |  
 Asian\_race == 1 |   
 Other\_race == 1 |  
 demo\_ethn\_v2 == 1, 0, 1),  
 Missing\_race = if\_else(is.na(Missing\_race) == T, 1, Missing\_race)) %>% # mark all NAs in missing race as 1  
 mutate(Indigenous\_race = if\_else(AIAN\_race == 1 | NHPI\_race == 1, 1, 0))   
   
  
abcd\_data <- abcd\_data %>%  
 mutate(pubertal\_status = if\_else(demo\_sex\_v2 == '1' | demo\_sex\_v2 == '3',  
 pds\_p\_ss\_male\_category\_2, # Set to this if the condition above is TRUE  
 pds\_p\_ss\_female\_category\_2)) # Otherwise set to this   
  
  
abcd\_data <- abcd\_data %>%  
 mutate(across(c(demo\_prnt\_ed\_v2\_comb, demo\_prtnr\_ed\_v2\_comb,   
 demo\_prnt\_ed\_v2\_bl, demo\_prtnr\_ed\_v2\_bl), ~as.integer(.x))) %>%  
 mutate(highest\_demo\_ed\_comb = case\_when(  
 is.na(demo\_prtnr\_ed\_v2\_comb) == T ~   
 demo\_prnt\_ed\_v2\_comb,  
 demo\_prnt\_ed\_v2\_comb > demo\_prtnr\_ed\_v2\_comb ~   
 demo\_prnt\_ed\_v2\_comb,  
 demo\_prnt\_ed\_v2\_comb <= demo\_prtnr\_ed\_v2\_comb ~   
 demo\_prtnr\_ed\_v2\_comb),  
 highest\_demo\_ed\_bl = case\_when(  
 is.na(demo\_prtnr\_ed\_v2\_bl) == T ~ demo\_prnt\_ed\_v2\_bl,  
 demo\_prnt\_ed\_v2\_bl > demo\_prtnr\_ed\_v2\_bl ~   
 demo\_prnt\_ed\_v2\_bl,  
 demo\_prnt\_ed\_v2\_bl <= demo\_prtnr\_ed\_v2\_bl ~   
 demo\_prtnr\_ed\_v2\_bl))  
  
  
abcd\_data <- abcd\_data %>%  
 # aggregate ksad questions to create meaningful, new variables   
 mutate(SI\_ever\_y = case\_when(  
 ksads\_23\_946\_t\_new + ksads\_23\_947\_t\_new + ksads\_23\_948\_t\_new + ksads\_23\_949\_t\_new + ksads\_23\_950\_t\_new > 0 ~ "present",  
 ksads\_23\_957\_t\_new + ksads\_23\_958\_t\_new + ksads\_23\_959\_t\_new + ksads\_23\_960\_t\_new + ksads\_23\_961\_t\_new > 0 ~ "past",  
 is.na(ksads\_23\_946\_t\_new) == T |is.na(ksads\_23\_947\_t\_new) == T |is.na(ksads\_23\_948\_t\_new) == T |   
 is.na(ksads\_23\_949\_t\_new) == T |is.na(ksads\_23\_950\_t\_new) == T |is.na(ksads\_23\_957\_t\_new) == T |  
 is.na(ksads\_23\_958\_t\_new) == T |is.na(ksads\_23\_959\_t\_new) == T |is.na(ksads\_23\_960\_t\_new) == T |  
 is.na(ksads\_23\_961\_t\_new) == T~ NA, # check if vars have NA and preserve the NAs   
 TRUE ~ "never"), # TRUE is case\_when's "else"   
 SI\_ever\_p = case\_when(   
 ksads\_23\_946\_p\_new + ksads\_23\_947\_p\_new + ksads\_23\_948\_p\_new + ksads\_23\_949\_p\_new + ksads\_23\_950\_p\_new > 0 ~ "present",   
 ksads\_23\_957\_p\_new + ksads\_23\_958\_p\_new + ksads\_23\_959\_p\_new + ksads\_23\_960\_p\_new + ksads\_23\_961\_p\_new > 0 ~ "past",  
 is.na(ksads\_23\_946\_p\_new) == T |is.na(ksads\_23\_947\_p\_new) == T |is.na(ksads\_23\_948\_p\_new) == T |  
 is.na(ksads\_23\_949\_p\_new) == T |is.na(ksads\_23\_950\_p\_new) == T |is.na(ksads\_23\_957\_p\_new) == T |  
 is.na(ksads\_23\_958\_p\_new) == T |is.na(ksads\_23\_959\_p\_new) == T |is.na(ksads\_23\_960\_p\_new) == T |  
 is.na(ksads\_23\_961\_p\_new) == T~ NA,  
 TRUE ~ "never"),  
 SI\_ever\_p\_y = case\_when(  
 SI\_ever\_p == "present" | SI\_ever\_y == "present" ~ "present",   
 SI\_ever\_p == "past" | SI\_ever\_y == "past" ~ "past",  
 SI\_ever\_p == "never" | SI\_ever\_y == "never" ~ "never"),  
 SA\_ever\_y = case\_when(  
 is.na(ksads\_23\_954\_t\_new) == T | is.na(ksads\_23\_965\_t\_new) == T ~ NA,  
 ksads\_23\_954\_t\_new > 0 ~ "present",   
 ksads\_23\_965\_t\_new > 0 ~ "past",  
 ksads\_23\_954\_t\_new <= 0 & ksads\_23\_965\_t\_new <= 0 ~ "never"),  
 SA\_ever\_p = case\_when(  
 ksads\_23\_954\_p\_new > 0 ~ "present",  
 ksads\_23\_965\_p\_new > 0 ~ "past",  
 ksads\_23\_954\_p\_new <= 0 & ksads\_23\_965\_p\_new <= 0 ~ "never"),  
 SA\_ever\_p\_y = case\_when(  
 SA\_ever\_p == "present" | SA\_ever\_y == "present" ~ "present",  
 SA\_ever\_p == "past" | SA\_ever\_y == "past" ~ "past",  
 SA\_ever\_p == "never" | SA\_ever\_y == "never" ~ "never"),  
 STB\_highest\_ever\_y = case\_when(  
 is.na(ksads\_23\_954\_t\_new) == T | is.na(ksads\_23\_965\_t\_new) == T |is.na(ksads\_23\_963\_t\_new) == T|   
 is.na(ksads\_23\_964\_t\_new) == T| is.na(ksads\_23\_953\_t\_new)== T | is.na(ksads\_23\_952\_t\_new) == T|  
 is.na(ksads\_23\_949\_t\_new) == T| is.na(ksads\_23\_960\_t\_new) == T|is.na(ksads\_23\_959\_t\_new) == T|  
 is.na(ksads\_23\_961\_t\_new) == T| is.na(ksads\_23\_948\_t\_new) == T| is.na(ksads\_23\_950\_t\_new) == T|  
 is.na(ksads\_23\_947\_t\_new) == T| is.na(ksads\_23\_958\_t\_new) == T| is.na(ksads\_23\_957\_t\_new) == T|  
 is.na(ksads\_23\_946\_t\_new)== T ~ NA,   
   
 0 < (ksads\_23\_954\_t\_new) + (ksads\_23\_965\_t\_new) ~ "SA",  
 0 < (ksads\_23\_963\_t\_new) + (ksads\_23\_964\_t\_new) + (ksads\_23\_953\_t\_new) + (ksads\_23\_952\_t\_new) ~ "SA interrup/aborted",  
 0 < (ksads\_23\_949\_t\_new) + (ksads\_23\_960\_t\_new) ~ "SI intent",  
 0 < (ksads\_23\_959\_t\_new) + (ksads\_23\_961\_t\_new) + (ksads\_23\_948\_t\_new) + (ksads\_23\_950\_t\_new) ~ "SI plan/method",  
 0 < (ksads\_23\_947\_t\_new) + (ksads\_23\_958\_t\_new) ~ "SI active nonspecific",  
 0 < (ksads\_23\_957\_t\_new) + (ksads\_23\_946\_t\_new) ~ "SI passive",  
 TRUE ~ "none"),  
 STB\_highest\_current\_y = case\_when(  
 is.na(ksads\_23\_954\_t\_new) == T | is.na(ksads\_23\_953\_t\_new) == T | is.na(ksads\_23\_952\_t\_new) == T|  
 is.na(ksads\_23\_949\_t\_new) == T| is.na(ksads\_23\_948\_t\_new) == T| is.na(ksads\_23\_950\_t\_new)== T |  
 is.na(ksads\_23\_947\_t\_new)== T | is.na(ksads\_23\_946\_t\_new)== T ~ NA,  
   
 0 < (ksads\_23\_954\_t\_new) ~ "SA",  
 0 < (ksads\_23\_953\_t\_new) + (ksads\_23\_952\_t\_new) ~ "SA interrup/aborted", # in original version of code this level is skipped (all participants in this level became NA)-- in new version it's preserved  
 0 < (ksads\_23\_949\_t\_new) ~ "SI intent",  
 0 < (ksads\_23\_948\_t\_new) + (ksads\_23\_950\_t\_new) ~ "SI plan/method",  
 0 < (ksads\_23\_947\_t\_new) ~ "SI active nonspecific",  
 0 < (ksads\_23\_946\_t\_new) ~ "SI passive",  
 TRUE ~ "none")) %>%  
 mutate(across(c(SI\_ever\_y, SI\_ever\_p,SI\_ever\_p\_y, SA\_ever\_y,SA\_ever\_p,SA\_ever\_p\_y, STB\_highest\_ever\_y,STB\_highest\_current\_y), ~as.factor(.x))) # convert all new vars to factors  
  
abcd\_data <- abcd\_data %>%  
 mutate(NSSI\_ever\_y = case\_when(  
 is.na(ksads\_23\_945\_t\_new) == T | is.na(ksads\_23\_956\_t\_new) == T ~ NA,  
 ksads\_23\_945\_t\_new > 0 ~ "present",  
 ksads\_23\_956\_t\_new > 0 ~ "past",  
 TRUE ~ "never"),  
 NSSI\_ever\_p = case\_when(  
 is.na(ksads\_23\_945\_p\_new) == T | is.na(ksads\_23\_956\_p\_new) == T ~ NA,  
 ksads\_23\_945\_p\_new > 0 ~ "present",  
 ksads\_23\_956\_p\_new > 0 ~ "past",   
 TRUE ~"never"),  
 NSSI\_ever\_p\_y = case\_when(  
 NSSI\_ever\_p == "present" | NSSI\_ever\_y == "present" ~ "present",  
 NSSI\_ever\_p == "past" | NSSI\_ever\_y == "past" ~ "past",  
 NSSI\_ever\_p == "never" | NSSI\_ever\_y == "never" ~ "never"),   
 SITB\_ever\_y = case\_when(  
 is.na(NSSI\_ever\_y) == T | is.na(STB\_highest\_ever\_y) == T ~ NA,  
 STB\_highest\_ever\_y != "none" | NSSI\_ever\_y != "never" ~ 1,  
 TRUE ~ 0)) %>%  
 mutate(across(c(NSSI\_ever\_y, NSSI\_ever\_p, NSSI\_ever\_p\_y, SITB\_ever\_y), ~as.factor(.x)))  
  
abcd\_data <- abcd\_data %>%  
 mutate(MDD\_ever\_y = case\_when(  
 is.na(ksads\_1\_840\_t\_new) == T |is.na(ksads\_1\_841\_t\_new) == T | is.na(ksads\_1\_842\_t\_new) == T ~ NA,  
 0 < (ksads\_1\_840\_t\_new) ~ "present",  
 0 < (ksads\_1\_841\_t\_new) ~ "partial remission",  
 0< (ksads\_1\_842\_t\_new) ~ "past",  
 TRUE ~ "never"),  
 AnyDD\_ever\_y = case\_when(  
 is.na(ksads\_1\_840\_t\_new) == T |is.na(ksads\_1\_841\_t\_new) == T | is.na(ksads\_1\_842\_t\_new) == T |  
 is.na(ksads\_1\_843\_t\_new) == T | is.na(ksads\_1\_844\_t\_new) == T |is.na(ksads\_1\_845\_t\_new) == T |   
 is.na(ksads\_1\_846\_t\_new) == T | is.na(ksads\_1\_847\_t\_new) == T ~ NA,  
 0 < (ksads\_1\_840\_t\_new) + (ksads\_1\_843\_t\_new) + (ksads\_1\_846\_t\_new) ~ "present",  
 0 < (ksads\_1\_841\_t\_new) + (ksads\_1\_844\_t\_new) ~ "partial remission",  
 0 < (ksads\_1\_842\_t\_new) + (ksads\_1\_845\_t\_new) + (ksads\_1\_847\_t\_new) ~ "past",  
 TRUE ~ "never"),  
 MDD\_ever\_p = case\_when(  
 is.na(ksads\_1\_840\_p\_new) == T |is.na(ksads\_1\_841\_p\_new) == T | is.na(ksads\_1\_842\_p\_new) == T ~ NA,  
 0 < (ksads\_1\_840\_p\_new) ~ "present",  
 0 < (ksads\_1\_841\_p\_new) ~ "partial remission",  
 0 < (ksads\_1\_842\_p\_new) ~ "past",  
 TRUE ~ "never"),  
 AnyDD\_ever\_p = case\_when(  
 is.na(ksads\_1\_840\_p\_new) == T |is.na(ksads\_1\_841\_p\_new) == T | is.na(ksads\_1\_842\_p\_new) == T |  
 is.na(ksads\_1\_843\_p\_new) == T | is.na(ksads\_1\_844\_p\_new) == T |is.na(ksads\_1\_845\_p\_new) == T |   
 is.na(ksads\_1\_846\_p\_new) == T | is.na(ksads\_1\_847\_p\_new) == T ~ NA,  
 0 < (ksads\_1\_840\_p\_new) + (ksads\_1\_843\_p\_new) + (ksads\_1\_846\_p\_new) ~ "present",  
 0 < (ksads\_1\_841\_p\_new) + (ksads\_1\_844\_p\_new) ~ "partial remission",  
 0 < (ksads\_1\_842\_p\_new) + (ksads\_1\_845\_p\_new) + (ksads\_1\_847\_p\_new) ~ "past",  
 TRUE ~ "never"),   
 MDD\_ever\_p\_y = case\_when(  
 MDD\_ever\_p == "present" | MDD\_ever\_y == "present" ~ "present",  
 MDD\_ever\_p == "partial remission" | MDD\_ever\_y == "partial remission" ~ "partial remission",  
 MDD\_ever\_p == "past" | MDD\_ever\_y == "past" ~ "past",  
 MDD\_ever\_p == "never" | MDD\_ever\_y == "never" ~ "never"),  
 AnyDD\_ever\_p\_y = case\_when(  
 AnyDD\_ever\_p == "present" | AnyDD\_ever\_y == "present" ~ "present",  
 AnyDD\_ever\_p == "partial remission" | AnyDD\_ever\_y == "partial remission" ~ "partial remission",  
 AnyDD\_ever\_p == "past" | AnyDD\_ever\_y == "past" ~ "past",  
 AnyDD\_ever\_p == "never" | AnyDD\_ever\_y == "never" ~ "never"  
 )) %>%  
 mutate(across(c(MDD\_ever\_y, AnyDD\_ever\_y, MDD\_ever\_p, AnyDD\_ever\_y, MDD\_ever\_p\_y, AnyDD\_ever\_p\_y), ~as.factor(.x)))  
  
abcd\_data <- abcd\_data %>%  
 select(-num\_range("demo\_race\_a\_p\_\_\_", 12:25), # race vars  
 -num\_range("ksads\_23\_", 946:950, "\_t\_new"), # SI, SA, NSSI vars  
 -num\_range("ksads\_23\_", 952:954, "\_t\_new"),  
 -num\_range("ksads\_23\_", 957:961, "\_t\_new"),  
 -num\_range("ksads\_23\_", 963:965, "\_t\_new"),  
 -num\_range("ksads\_23\_", 946:950, "\_p\_new"),  
 -ksads\_23\_954\_p\_new,  
 -num\_range("ksads\_23\_", 957:961, "\_p\_new"),   
 -num\_range("ksads\_1\_", 840:847, "\_t\_new"), # depression vars  
 -num\_range("ksads\_1\_", 840:847, "\_p\_new"))  
  
abcd\_data.selected\_time <- abcd\_data %>%  
 filter(eventname %in% timepoint\_list)  
# Creates name based on date and initials/string passed in by user. If no date given, use the current date  
if(is.null(out\_date)){  
 out\_date <- Sys.Date()  
}  
# If no initials/string is given, throw an error that user must input one  
if(is.null(out\_initials)){  
 stop("No input given for 'out\_initials'.   
 User must provide string input to create output file name.")  
}  
# Now output file based on name and out\_dir given. Recall that if out\_dir is NULL, it will write to current wd  
csv\_out\_name <- paste0(out\_dir,'/',out\_date,'\_',out\_initials,'.csv')  
write.csv(abcd\_data.selected\_time, csv\_out\_name, row.names = FALSE)   
colnames(abcd\_data.selected\_time)