Script for Manuscript: The bidirectional relationship between brain features and the dysregulation profile: a longitudinal, multi-modal approach

Elisabet Blok, Sander Lamballais, Laia Benitez-Manzanas, Tonya White

2 May 2022

Contents

1. Load required libraries

```
library(foreign) #used to load data
library(mice) #used for imputations
library(lavaan) #used to fit (RI-)CLPMs
library(survey) #used to account for family structure
library(lavaan.survey) #used to account for family structure
library(mitools) #used to allow imputed dataset analysis in (RI-)CLPMs
library(meta) #used for meta-analysis
library(kableExtra) #used to make tables
library(stringi) #used for general formatting
library(stringr) #used for general formatting
library(tidyr) #used for general formatting
library(dplyr) #used for general formatting
```

2. Primary exploratory analyses

2.1. Generation R

2.1.1. Load data

```
#Set working directory to load data
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/2.Data")

##MRI data
#T1-weighted
#Cortex
mri6_aparc_genr <- readRDS("f05_freesurfer_v6_24june2021_aparc_stats_pull18Aug2021.rds")
mri6_aparc_genr <- mri6_aparc_genr %>% rename(
  IDC = idc
)
mri9_aparc_genr <- readRDS("f09_freesurfer_v6_09dec2016_aparc_stats_pull06june2017.rds")
mri9_aparc_genr <- mri9_aparc_genr %>% rename(
  IDC = idc
)
mri13_aparc_genr <- readRDS("f13_freesurfer_14oct2020_aparc_stats_pull23Nov2020_noDups.rds")

#Subcortex
mri6_aseg_genr <- readRDS("f05_freesurfer_v6_24june2021_aseg_stats_pull18Aug2021.rds")
mri6_aseg_genr <- mri6_aseg_genr %>% rename(
  IDC = idc
)
mri9_aseg_genr <- readRDS("f09_freesurfer_v6_09dec2016_aseg_stats_pull06june2017.rds")
mri9_aseg_genr <- mri9_aseg_genr %>% rename(
  IDC = idc
)
mri13_aseg_genr <- readRDS("f13_freesurfer_14oct2020_aseg_stats_pull23Nov2020_noDups.rds")
```

```r
#DTI
dti6_genr <- readRDS("f05_GenR_MRI_eddy_dipy_wls_14Feb2022_autoPtx_dti_stats_v1.rds")
dti9_genr <- readRDS("f09_GenR_MRI_eddy_dipy_wls_14Feb2022_autoPtx_dti_stats_v1.rds")
dti13_genr <- readRDS("f13_GenR_MRI_eddy_dipy_wls_14Feb2022_autoPtx_dti_stats_v1.rds")

dti6_genr <- rename(dti6_genr, IDC = idc)
dti9_genr <- rename(dti9_genr, IDC = idc)
dti13_genr <- rename(dti13_genr, IDC = idc)

dti6_genr[,2:379] <- lapply(dti6_genr[,2:379], as.numeric)
dti9_genr[,2:379] <- lapply(dti9_genr[,2:379], as.numeric)
dti13_genr[,2:379] <- lapply(dti13_genr[,2:379], as.numeric)

#Core data scan usability
#Also includes covariate age at MRI
#T1w images
core_genr <- readRDS("genr_mri_core_data_20220311.rds")
core_genr <- core_genr %>% rename(
  IDC = idc
)

##CBCL data
#Also includes covariate age at CBCL
cbcl6_genr <- read.spss("CBCL_6_incl_Tscores_20201111.sav", to.data.frame = T)
cbcl9_genr <- read.spss("CBCL_9_incl_Tscores_20201111.sav", to.data.frame = T)
cbcl13_genr <- read.spss("GR1093-E1_CBCL_18062020.sav", to.data.frame = T)

##Covariates
#General data, includes biological sex, child national origin and maternal education
general_genr <- read.spss("CHILD-ALLGENERALDATA_29012018.sav", to.data.frame = T)

#Handedness
handedness6_genr <- read.spss("MRI5HANDEDNESS_21082013.sav", to.data.frame = T)
handedness9_genr <- read.spss("CHILD_Edingburgh HandednessF9_30102020.sav", to.data.frame = T)
handedness13_genr <-
  read.spss("CHILDMRI13_Gr1098_C1_EdingburghHandedness_26022021.sav", to.data.frame = T)

#ICV
mri6_icv_genr <- readRDS("f05_freesurfer_v6_24june2021_tbv_stats_pull18Aug2021.rds")
mri6_icv_genr <- mri6_icv_genr %>% rename(
  IDC = idc
)
mri9_icv_genr <- readRDS("f09_freesurfer_v6_09dec2016_tbv_stats_pull20june2017.rds")
mri9_icv_genr <- mri9_icv_genr %>% rename(
  IDC = idc
)
mri13_icv_genr <- readRDS("f13_freesurfer_14oct2020_tbv_stats_pull23Nov2020_noDups.rds")

##Data for sensitivity analyses
#Child cognitive performance
cog_perf_genr <- read.spss("CHILDWISC13_16082021.sav", to.data.frame = T)

#Child medication use
```

```
meduse_genr <- read.spss("20220217_Elisabeth_PsychotropicMedication13.sav", to.data.frame = T)
```

2.1.2. Select variables and merge data

```
##Select variables
##MRI data
#Cortex
mri6_aparc_sel_genr <- select(mri6_aparc_genr, c(IDC, ends_with("_vol_f05")))
mri9_aparc_sel_genr <- select(mri9_aparc_genr, c(IDC, ends_with("_vol_f09")))
mri13_aparc_sel_genr <- select(mri13_aparc_genr, c(IDC, ends_with("_vol_f13")))

#Subcortical structures
mri6_aseg_sel_genr <-
  select(mri6_aseg_genr, c(IDC, Left_Thalamus_Proper_vol_f05, Left_Caudate_vol_f05,
                       Left_Putamen_vol_f05, Left_Pallidum_vol_f05,
                       Left_Hippocampus_vol_f05, Left_Amygdala_vol_f05,
                       Left_Accumbens_area_vol_f05,Right_Thalamus_Proper_vol_f05,
                       Right_Caudate_vol_f05, Right_Putamen_vol_f05,
                       Right_Pallidum_vol_f05, Right_Hippocampus_vol_f05,
                       Right_Amygdala_vol_f05, Right_Accumbens_area_vol_f05))
mri9_aseg_sel_genr <-
  select(mri9_aseg_genr, c(IDC, Left_Thalamus_Proper_vol_f09, Left_Caudate_vol_f09,
                       Left_Putamen_vol_f09, Left_Pallidum_vol_f09,
                       Left_Hippocampus_vol_f09, Left_Amygdala_vol_f09,
                       Left_Accumbens_area_vol_f09,Right_Thalamus_Proper_vol_f09,
                       Right_Caudate_vol_f09, Right_Putamen_vol_f09,
                       Right_Pallidum_vol_f09, Right_Hippocampus_vol_f09,
                       Right_Amygdala_vol_f09, Right_Accumbens_area_vol_f09))
mri13_aseg_sel_genr <-
  select(mri13_aseg_genr, c(IDC, Left_Thalamus_Proper_vol_f13, Left_Caudate_vol_f13,
                       Left_Putamen_vol_f13, Left_Pallidum_vol_f13,
                       Left_Hippocampus_vol_f13, Left_Amygdala_vol_f13,
                       Left_Accumbens_area_vol_f13,Right_Thalamus_Proper_vol_f13,
                       Right_Caudate_vol_f13, Right_Putamen_vol_f13,
                       Right_Pallidum_vol_f13, Right_Hippocampus_vol_f13,
                       Right_Amygdala_vol_f13, Right_Accumbens_area_vol_f13))

#DTI data
dti6_sel_genr <-
  select(dti6_genr, c(IDC, ends_with("_wls_wavg_FA_f05"), ends_with("_wls_wavg_MD_f05")))
dti9_sel_genr <-
  select(dti9_genr, c(IDC, ends_with("_wls_wavg_FA_f09"), ends_with("_wls_wavg_MD_f09")))
dti13_sel_genr <-
  select(dti13_genr, c(IDC, ends_with("_wls_wavg_FA_f13"), ends_with("_wls_wavg_MD_f13")))

#Core data scan usability
core_sel_genr <- select(core_genr, c(IDC, mri_consent_f05, mri_consent_f09, mri_consent_f13,
                           age_child_mri_f05, age_child_mri_f09, age_child_mri_f13,
                           t1_has_nii_f05, t1_has_nii_f09, t1_asset_has_nii_f09,
                           t1_has_nii_f13, has_braces_mri_f05, has_braces_mri_f09,
                           has_braces_mri_f13, exclude_incidental_f05,
                           exclude_incidental_f09, exclude_incidental_f13,
                           freesurfer_qc_f05, freesurfer_qc_f09, freesurfer_qc_f13,
```

```
                           dti_has_nii_f05, dti_has_nii_f09, dti_has_nii_f13,
                           dti_overall_qc_f05, dti_overall_qc_f09, dti_overall_qc_f13))
##CBCL data
cbcl6_sel_genr <- select(cbcl6_genr, c(IDC, agechild_GR1075, sum_emo_5, sum_anx_5, sum_som_5,
                             sum_wid_5, sum_sle_5, sum_att_5, sum_agg_5, cbcl_sum_5))
cbcl9_sel_genr <- select(cbcl9_genr, c(IDC, AgeChild_CBCL9m, sum_anx_9m, sum_wit_9m,
                             sum_som_9m, sum_sop_9m, sum_tho_9m, sum_att_9m,
                             sum_rul_9m, sum_agg_9m, cbcl_sum_9m))
cbcl13_sel_genr <- select(cbcl13_genr, c(IDC, AGECHILD_GR1093, sum_anx_14, sum_wit_14,
                             sum_som_14, sum_sop_14, sum_tho_14, sum_att_14,
                             sum_rul_14, sum_agg_14, cbcl_sum_14))

##Covariates
#General
general_sel_genr <- select(general_genr, c(IDC, IDM, GENDER, EDUCM5, ETHNINFv2))

#Handedness
handedness6_sel_genr <- select(handedness6_genr, c(IDC, HC12))
handedness9_sel_genr <- select(handedness9_genr, c(IDC, HC12))
handedness13_sel_genr <- select(handedness13_genr, c(IDC, HC12_F13))

#Rename handedness variable F5/F9 to make it consistent with other variable namings
colnames(handedness6_sel_genr) <- c("IDC", "HC12_F5")
colnames(handedness9_sel_genr) <- c("IDC", "HC12_F9")

#ICV
mri6_icv_sel_genr <- select(mri6_icv_genr, c(IDC, eTIV_f05))
mri9_icv_sel_genr <- select(mri9_icv_genr, c(IDC, eTIV_f09))
mri13_icv_sel_genr <- select(mri13_icv_genr, c(IDC, eTIV_f13))

##Data for sensitivity analyses
#Child cognitive performance
cog_perf_sel_genr <- select(cog_perf_genr, c(IDC, WISC13_FSIQ))

#Child medication use
meduse_sel_genr <- select(meduse_genr, c(IDC, psychotropic_meduse13))

#Merge data
df1_genr <- merge(general_sel_genr, mri6_icv_sel_genr, by = "IDC", all = T)
df2_genr <- merge(df1_genr, mri9_icv_sel_genr, by = "IDC", all = T)
df3_genr <- merge(df2_genr, mri13_icv_sel_genr, by = "IDC", all = T)
df4_genr <- merge(df3_genr, core_sel_genr, by = "IDC", all = T)
df5_genr <- merge(df4_genr, handedness6_sel_genr, by = "IDC", all = T)
df6_genr <- merge(df5_genr, handedness9_sel_genr, by = "IDC", all = T)
df7_genr <- merge(df6_genr, handedness13_sel_genr, by = "IDC", all = T)
df8_genr <- merge(df7_genr, cbcl6_sel_genr, by = "IDC", all = T)
df9_genr <- merge(df8_genr, cbcl9_sel_genr, by = "IDC", all = T)
df10_genr <- merge(df9_genr, cbcl13_sel_genr, by = "IDC", all = T)
df11_genr <- merge(df10_genr, mri6_aseg_sel_genr, by = "IDC", all = T)
df12_genr <- merge(df11_genr, mri6_aparc_sel_genr, by = "IDC", all = T)
df13_genr <- merge(df12_genr, mri9_aseg_sel_genr, by = "IDC", all = T)
df14_genr <- merge(df13_genr, mri9_aparc_sel_genr, by = "IDC", all = T)
df15_genr <- merge(df14_genr, mri13_aseg_sel_genr, by = "IDC", all = T)
```

```
df16_genr <- merge(df15_genr, mri13_aparc_sel_genr, by = "IDC", all = T)
df17_genr <- merge(df16_genr, dti6_sel_genr, by = "IDC", all = T)
df18_genr <- merge(df17_genr, dti9_sel_genr, by = "IDC", all = T)
df19_genr <- merge(df18_genr, dti13_sel_genr, by = "IDC", all = T)
df20_genr <- merge(df19_genr, cog_perf_sel_genr, by = "IDC", all = T)
df_genr <- merge(df20_genr, meduse_sel_genr, by = "IDC", all = T)
```

2.1.3. Select participants

```
#Create variable that specifies whether or not to use data from that age for MRI data
#T1w
df_genr$avail_f5 <-
  ifelse((df_genr$mri_consent_f05 == "yes" & df_genr$t1_has_nii_f05 == "yes" &
            df_genr$has_braces_mri_f05 == "no" & df_genr$exclude_incidental_f05 == "include" &
            df_genr$freesurfer_qc_f05 == "usable"), 1, 0)
df_genr$avail_f9 <-
  ifelse((df_genr$mri_consent_f09 == "yes" & df_genr$t1_has_nii_f09 == "yes" &
            df_genr$t1_asset_has_nii_f09 != "exclude" &
            df_genr$has_braces_mri_f09 == "no" & df_genr$exclude_incidental_f09 == "include" &
            df_genr$freesurfer_qc_f09 == "usable"), 1, 0)
df_genr$avail_f13 <-
  ifelse((df_genr$mri_consent_f13 == "yes" & df_genr$t1_has_nii_f13 == "yes" &
            df_genr$has_braces_mri_f13 == "no" & df_genr$exclude_incidental_f13 == "include" &
            df_genr$freesurfer_qc_f13 == "usable"), 1, 0)
#Rename NA to no
df_genr$avail_f5 <- ifelse(is.na(df_genr$avail_f5), 0,
                                  ifelse(df_genr$avail_f5 == 0, 0, 1))
df_genr$avail_f9 <- ifelse(is.na(df_genr$avail_f9), 0,
                                   ifelse(df_genr$avail_f9 == 0, 0, 1))
df_genr$avail_f13 <- ifelse(is.na(df_genr$avail_f13), 0,
                                   ifelse(df_genr$avail_f13 == 0, 0, 1))

#DTI
#QC variable is coded unusable or NA, which is why we select all those with NA here
df_genr$avail_f5_dti <-
  ifelse(!is.na(rowSums(df_genr[,312:365])) &
    (df_genr$mri_consent_f05 == "yes" &
              df_genr$dti_has_nii_f05 == "yes" & is.na(df_genr$dti_overall_qc_f05) &
              df_genr$has_braces_mri_f05 == "no" & df_genr$exclude_incidental_f05 == "include"), 1, 0)
df_genr$avail_f9_dti <-
  ifelse(!is.na(rowSums(df_genr[,366:419])) &
    (df_genr$mri_consent_f09 == "yes" &
              df_genr$dti_has_nii_f09 == "yes" & is.na(df_genr$dti_overall_qc_f09) &
              df_genr$has_braces_mri_f09 == "no" & df_genr$exclude_incidental_f09 == "include"), 1, 0)
df_genr$avail_f13_dti <-
  ifelse(!is.na(rowSums(df_genr[,420:473])) &
    (df_genr$mri_consent_f13 == "yes" &
              df_genr$dti_has_nii_f13 == "yes" & is.na(df_genr$dti_overall_qc_f13) &
              df_genr$has_braces_mri_f13 == "no" & df_genr$exclude_incidental_f13 == "include"), 1, 0)

#Rename NA to no
df_genr$avail_f5_dti <- ifelse(is.na(df_genr$avail_f5_dti), 0,
                                  ifelse(df_genr$avail_f5_dti == 0, 0, 1))
```

```
df_genr$avail_f9_dti <- ifelse(is.na(df_genr$avail_f9_dti), 0,
                               ifelse(df_genr$avail_f9_dti == 0, 0, 1))
df_genr$avail_f13_dti <- ifelse(is.na(df_genr$avail_f13_dti), 0,
                               ifelse(df_genr$avail_f13_dti == 0, 0, 1))


#Create column that specifies how many scans are available
df_genr$nscans <- rowSums(df_genr[,c("avail_f5", "avail_f9", "avail_f13")])
df_genr$nscans_dti <- rowSums(df_genr[,c("avail_f5_dti", "avail_f9_dti", "avail_f13_dti")])

#Create column that specifies how many CBCL data is available
df_genr$ncbcl <-
  ifelse(!is.na(df_genr$cbcl_sum_5) & !is.na(df_genr$cbcl_sum_9m) & !is.na(df_genr$cbcl_sum_14), 3,
         ifelse(!is.na(df_genr$cbcl_sum_5) & !is.na(df_genr$cbcl_sum_9m) &
                is.na(df_genr$cbcl_sum_14) |
                  !is.na(df_genr$cbcl_sum_5) & is.na(df_genr$cbcl_sum_9m) &
                !is.na(df_genr$cbcl_sum_14) |
                  is.na(df_genr$cbcl_sum_5) & !is.na(df_genr$cbcl_sum_9m) & !is.na(df_genr$cbcl_sum_14)
                ifelse(is.na(df_genr$cbcl_sum_5) & is.na(df_genr$cbcl_sum_9m) &
                is.na(df_genr$cbcl_sum_14),
                       0, 1)))

#Select those with cbcl & mri data on at least two time points
df_sel_genr <- subset(df_genr, (nscans > 1 | nscans_dti > 1) & ncbcl > 1)

#Set MRI data to missing if scans are not usable per age
#This needs to be done as we include everyone with at least 2 time-points
#but we do not want to include unusable scans
for(x in 1:nrow(df_sel_genr)){
  #T1w
  #F5
  if(df_sel_genr[x,"avail_f5"] == 0){
    df_sel_genr[x,c(6,66:147)] <- NA
  }
  #F9
  if(df_sel_genr[x,"avail_f9"] == 0){
    df_sel_genr[x,c(7,148:229)] <- NA
  }
  #F13
    if(df_sel_genr[x,"avail_f13"] == 0){
    df_sel_genr[x,c(8,230:311)] <- NA
    }
  #DTI
    #F5
  if(df_sel_genr[x,"avail_f5_dti"] == 0){
    df_sel_genr[x,c(312:365)] <- NA
  }
  #F9
  if(df_sel_genr[x,"avail_f9_dti"] == 0){
    df_sel_genr[x,c(366:419)] <- NA
  }
  #F13
    if(df_sel_genr[x,"avail_f13_dti"] == 0){
```

```
    df_sel_genr[x,c(420:473)] <- NA
    }
}

#Remove variables to select participants from dataset as they are no longer needed
df_com_genr <- select(df_sel_genr, -c(all_of(colnames(core_sel_genr[c(2:4,8:26)]))))
```

2.1.4. Transformations

```
##MRI data
#Combine hemispheres by calculating average of left & right hemispheres
#Create vector of all structure names
#Since the structure names are equal in all MRI datasets,
#we can just create a vector of structure names based on 1 measurement occasion
#Since all structures are included in the dataset twice (once for both hemispheres),
#we only take those belonging to the left hemisphere
aseg_cols_genr <- colnames(select(mri6_aseg_sel_genr, starts_with("Left_")))
aparc_cols_genr <- colnames(select(mri6_aparc_sel_genr, starts_with("lh_")))

#Remove hemisphere prefix
aseg_structs_temp_genr <- stri_remove_empty(unlist(strsplit(aseg_cols_genr, "Left_")))
aparc_structs_temp_genr <- stri_remove_empty(unlist(strsplit(aparc_cols_genr, "lh_")))

#Drop _f05 suffix
aseg_structs_genr <- stri_remove_empty(unlist(strsplit(aseg_structs_temp_genr, "_f05")))
aparc_structs_genr <- stri_remove_empty(unlist(strsplit(aparc_structs_temp_genr, "_f05")))

#DTI
#Select tracts of interest (just the left hemisphere because we'll drop hemisphere now)
dti_cols_genr <-
  colnames(select(dti6_sel_genr, (starts_with("cgc_l") | starts_with("cst_l") |
                    starts_with("unc_l") | starts_with("ilf_l") |
                    starts_with("slf_l")) & ends_with("_FA_f05")))
dti_tracts_genr <-
  stri_remove_empty(unlist(strsplit(dti_cols_genr, "_l_")))[seq(1, length(dti_cols_genr)*2, 2)]

#Combine aseg, aparc and dti vectors into one vector to loop over all of them in the next step
structs_genr <- c(aseg_structs_genr, aparc_structs_genr, dti_tracts_genr)

#Remove amygdala and nucleus accumbens volumes,
#as we analyze those structures for hemisphere specific effects
structs2_genr <- structs_genr[! structs_genr %in% c("Amygdala_vol", "Accumbens_area_vol")]

#Loop over all structures and average columns of the same structure in both hemispheres
#(at the same measurement occasion)
#Go over all structures
for(x in structs2_genr){
  #Go over all columns to identify the first column (left hemisphere)
  for(y in 1:ncol(df_com_genr)){
    #Store the column name to find the counterpart in the other (right) hemisphere
    a <- colnames(df_com_genr[y])
    #Go over all columns to identify the second column (right hemisphere)
    for(z in 1:ncol(df_com_genr)){
```

```
    #Store the column name to identify the same structures in each hemisphere
    b <- colnames(df_com_genr[z])
    #If we have the same structure for both hemispheres, calculate the mean of those

    ##MRI @6
    #Subcortical volumes MRI @6
    if(a == paste0("Left_",x,"_f05") & b == paste0("Right_",x,"_f05")){
      newcolname <- paste0(x,"_f05")
      df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
    }
    #Regional volumes MRI @6
    if(a == paste0("lh_",x,"_f05") & b == paste0("rh_",x,"_f05")){
      newcolname <- paste0(x,"_f05")
      df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
    }
    #DTI @6
    #FA
    if(a == paste0(x,"_l_dti_dipy_wls_wavg_FA_f05") & b == paste0(x,"_r_dti_dipy_wls_wavg_FA_f05")){
      newcolname <- paste0(x,"_FA_f05")
      df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
    }
    #MD
    if(a == paste0(x,"_l_dti_dipy_wls_wavg_MD_f05") & b == paste0(x,"_r_dti_dipy_wls_wavg_MD_f05")){
      newcolname <- paste0(x,"_MD_f05")
      df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
    }

    ##MRI @9
    #Subcortical volumes MRI @9
    if(a == paste0("Left_",x,"_f09") & b == paste0("Right_",x,"_f09")){
      newcolname <- paste0(x,"_f09")
      df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
    }
    #Regional volumes MRI @9
    if(a == paste0("lh_",x,"_f09") & b == paste0("rh_",x,"_f09")){
      newcolname <- paste0(x,"_f09")
      df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
    }
    #DTI @9
    #FA
    if(a == paste0(x,"_l_dti_dipy_wls_wavg_FA_f09") & b == paste0(x,"_r_dti_dipy_wls_wavg_FA_f09")){
      newcolname <- paste0(x,"_FA_f09")
      df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
    }
    #MD
    if(a == paste0(x,"_l_dti_dipy_wls_wavg_MD_f09") & b == paste0(x,"_r_dti_dipy_wls_wavg_MD_f09")){
      newcolname <- paste0(x,"_MD_f09")
      df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
    }

    ##MRI @13
    #Subcortical volumes MRI @13
    if(a == paste0("Left_",x,"_f13") & b == paste0("Right_",x,"_f13")){
```

```
        newcolname <- paste0(x,"_f13")
        df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
      }
      #Regional volumes MRI @13
      if(a == paste0("lh_",x,"_f13") & b == paste0("rh_",x,"_f13")){
        newcolname <- paste0(x,"_f13")
        df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
      }
      #DTI @9
      #FA
      if(a == paste0(x,"_l_dti_dipy_wls_wavg_FA_f13") & b == paste0(x,"_r_dti_dipy_wls_wavg_FA_f13")){
        newcolname <- paste0(x,"_FA_f13")
        df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
      }
      #MD
      if(a == paste0(x,"_l_dti_dipy_wls_wavg_MD_f13") & b == paste0(x,"_r_dti_dipy_wls_wavg_MD_f13")){
        newcolname <- paste0(x,"_MD_f13")
        df_com_genr[,newcolname] <- (df_com_genr[,y] + df_com_genr[,z])/2
      }
    }
  }
}

#Now tidy up the dataset
#Remove all original hemisphere specific variables
df_tidy_genr <-
  select(df_com_genr,
         -c(starts_with(c(unlist(lapply(structs2_genr[1:5], function(x) paste0("Left_",x))),
                          unlist(lapply(structs2_genr[1:5], function(x) paste0("Right_",x))),
                          unlist(lapply(structs2_genr[6:39], function(x) paste0("lh_",x))),
                          unlist(lapply(structs2_genr[6:39], function(x) paste0("rh_",x))))),
            ends_with("l_dti_dipy_wls_wavg_FA_f05"), ends_with("r_dti_dipy_wls_wavg_FA_f05"),
            ends_with("l_dti_dipy_wls_wavg_FA_f09"), ends_with("r_dti_dipy_wls_wavg_FA_f09"),
            ends_with("l_dti_dipy_wls_wavg_FA_f13"), ends_with("r_dti_dipy_wls_wavg_FA_f13"),
            ends_with("l_dti_dipy_wls_wavg_MD_f05"), ends_with("r_dti_dipy_wls_wavg_MD_f05"),
            ends_with("l_dti_dipy_wls_wavg_MD_f09"), ends_with("r_dti_dipy_wls_wavg_MD_f09"),
            ends_with("l_dti_dipy_wls_wavg_MD_f13"), ends_with("r_dti_dipy_wls_wavg_MD_f13"),
            starts_with("mcp")))

#Normalize to mean 0 and SD 1
mri_vars_genr <- colnames(df_tidy_genr)[c(6:8, 44:67, 79:225)]
for(x in mri_vars_genr){
  newcolname <- paste0(x, "_scaled")
  df_tidy_genr[,newcolname] <- c(scale(df_tidy_genr[,x]))
}

##CBCL data
df_tidy_genr$sum_dp_5 <- df_tidy_genr$sum_anx_5 + df_tidy_genr$sum_att_5 + df_tidy_genr$sum_agg_5
df_tidy_genr$sum_dp_9m <- df_tidy_genr$sum_anx_9m + df_tidy_genr$sum_att_9m + df_tidy_genr$sum_agg_9m
df_tidy_genr$sum_dp_14 <- df_tidy_genr$sum_anx_14 + df_tidy_genr$sum_att_14 + df_tidy_genr$sum_agg_14

#Sqrt transform & normalize to mean 0 and SD 1
cbcl_vars_genr <- colnames(df_tidy_genr)[c(16:23, 25:33, 35:43, 400:402)]
```

```r
for(x in cbcl_vars_genr){
  newcolname <- paste0(x, "_sqrt_scaled")
  df_tidy_genr[,newcolname] <- c(scale(sqrt(df_tidy_genr[,x])))
}

##Covariates
#For the CBCL at 5 age is in months, recalculate to years
df_tidy_genr$agechild_GR1075 <- df_tidy_genr$agechild_GR1075/12
#Maternal education
df_tidy_genr$maternal_education <-
  as.factor(ifelse(df_tidy_genr$EDUCM5 == "no education finished" |
                     df_tidy_genr$EDUCM5 == "primary", "low",
                   ifelse(df_tidy_genr$EDUCM5 == "secondary, phase 1" |
                            df_tidy_genr$EDUCM5 == "secondary, phase 2", "middle", "high")))

#Child national origin
df_tidy_genr$child_nationalorigin <-
  as.factor(ifelse(df_tidy_genr$ETHNINFv2 == "Dutch", "Dutch",
                   ifelse(df_tidy_genr$ETHNINFv2 == "American,western" |
                            df_tidy_genr$ETHNINFv2 == "Asian, western" |
                            df_tidy_genr$ETHNINFv2 == "European" |
                            df_tidy_genr$ETHNINFv2 == "Indonesian" |
                            df_tidy_genr$ETHNINFv2 == "Oceanie", "other western", "non western")))

#Psychotropic medication use
df_tidy_genr$meduse <- as.factor(ifelse(df_tidy_genr$psychotropic_meduse13 == "no", 0, 1))

#Rename forceps minor/forceps minor variables to be consistent with other DTI vars
df_tidy_genr2 <- df_tidy_genr %>%
  rename(
    fmi_FA_f05_scaled = fmi_dti_dipy_wls_wavg_FA_f05_scaled,
    fmi_FA_f09_scaled = fmi_dti_dipy_wls_wavg_FA_f09_scaled,
    fmi_FA_f13_scaled = fmi_dti_dipy_wls_wavg_FA_f13_scaled,
    fmi_MD_f05_scaled = fmi_dti_dipy_wls_wavg_MD_f05_scaled,
    fmi_MD_f09_scaled = fmi_dti_dipy_wls_wavg_MD_f09_scaled,
    fmi_MD_f13_scaled = fmi_dti_dipy_wls_wavg_MD_f13_scaled,
    fma_FA_f05_scaled = fma_dti_dipy_wls_wavg_FA_f05_scaled,
    fma_FA_f09_scaled = fma_dti_dipy_wls_wavg_FA_f09_scaled,
    fma_FA_f13_scaled = fma_dti_dipy_wls_wavg_FA_f13_scaled,
    fma_MD_f05_scaled = fma_dti_dipy_wls_wavg_MD_f05_scaled,
    fma_MD_f09_scaled = fma_dti_dipy_wls_wavg_MD_f09_scaled,
    fma_MD_f13_scaled = fma_dti_dipy_wls_wavg_MD_f13_scaled
  )

#Now tidy up the dataset
#Remove all unscaled variables
#Remove EDUCM5 & ETHNINFv2 since they have been recategorized
df_final_genr <-
  select(df_tidy_genr2,
         -c(mri_vars_genr, cbcl_vars_genr, EDUCM5, ETHNINFv2, psychotropic_meduse13))
```

2.1.5. Imputations

```
meth_genr <- make.method(df_final_genr)
#don't impute ID numbers, CBCL or MRI variables
meth_genr[c(1:2,14:225,228)] <- ""
qpredR_genr <- quickpred(df_final_genr)
#ID numbers, CBCL or MRI variables not as predictors
qpredR_genr[,c(1:2,14:225,228)] <- 0
#apply default predictor matrix rules
diag(qpredR_genr) <- 0; qpredR_genr[which(meth_genr == ""),] <- 0
#n predictors per imputed variable
rowSums(qpredR_genr); mean(rowSums(qpredR_genr[-which(rowSums(qpredR_genr)==0),]))
#test run
ini_genr <- mice(df_final_genr, predictorMatrix = qpredR_genr,
    maxit=1, m=1, printFlag=F, method = meth_genr)
#test logged events
ini_genr$loggedEvents

dsImp_genr <- mice(df_final_genr, predictorMatrix = qpredR_genr,
    maxit=30, m=30, method = meth_genr, seed = 2021)

#Save imputed dataframe
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/2.Data")
saveRDS(dsImp_genr, "dsImp_GenR.rds")
```

2.1.6. Cross-lagged panel model

```
##Make list of imputed datasets to feed to lavaan
implist_genr <- lapply(seq(dsImp_genr$m), function(im) complete(dsImp_genr, im))

##Create dummy variables for categorical variables
for(x in seq(dsImp_genr$m)){
  #Sex
  implist_genr[[x]]$sex <-
    ifelse(implist_genr[[x]]$GENDER == "girl", 1, 0)
  #Maternal education
  implist_genr[[x]]$maternal_education_high <-
    ifelse(implist_genr[[x]]$maternal_education == "high", 1, 0)
  implist_genr[[x]]$maternal_education_middle <-
    ifelse(implist_genr[[x]]$maternal_education == "middle", 1, 0)
  #Child national origin
  implist_genr[[x]]$child_nationalorigin_dutch <-
    ifelse(implist_genr[[x]]$child_nationalorigin == "Dutch", 1, 0)
  implist_genr[[x]]$child_nationalorigin_wes <-
    ifelse(implist_genr[[x]]$child_nationalorigin == "other western", 1, 0)
}

#Split T1-weighted and DTI sample
#Do this so that there are no missings in each dataframe
implist_t1w_genr <- list()
implist_dti_genr <- list()

for(x in seq(dsImp_genr$m)){
```

```
  implist_t1w_genr[[x]] <- subset(implist_genr[[x]],
                                  implist_genr[[x]]$nscans > 1)
}
for(x in seq(dsImp_genr$m)){
  implist_dti_genr[[x]] <- subset(implist_genr[[x]],
                                  implist_genr[[x]]$nscans_dti > 1)
}

#Create imputationlist object
data_genr_t1w <- imputationList(implist_t1w_genr)
data_genr_dti <- imputationList(implist_dti_genr)

#Loop over all MRI measures
#We reuse the vector structs and substitute this with our hemisphere specific ROIs
structs3_genr <- c(structs2_genr[1:39],
              "Left_Amygdala_vol", "Right_Amygdala_vol",
              "Left_Accumbens_area_vol", "Right_Accumbens_area_vol",
              sapply(structs2_genr[40:44], paste0, "_FA"),
              "fma_FA", "fmi_FA",
              sapply(structs2_genr[40:44], paste0, "_MD"),
              "fma_MD", "fmi_MD")

#Create empty dataframes to store results
results_CLPM_genr_m1_t1w <- data.frame()
fitmeaures_CLPM_genr_m1_t1w <- data.frame()
rsquared_CLPM_genr_m1_t1w <- data.frame()
results_CLPM_genr_m2_t1w <- data.frame()
fitmeaures_CLPM_genr_m2_t1w <- data.frame()
rsquared_CLPM_genr_m2_t1w <- data.frame()
results_CLPM_genr_m3_t1w <- data.frame()
fitmeaures_CLPM_genr_m3_t1w <- data.frame()
rsquared_CLPM_genr_m3_t1w <- data.frame()

results_CLPM_genr_m1_dti <- data.frame()
fitmeaures_CLPM_genr_m1_dti <- data.frame()
rsquared_CLPM_genr_m1_dti <- data.frame()
results_CLPM_genr_m2_dti <- data.frame()
fitmeaures_CLPM_genr_m2_dti <- data.frame()
rsquared_CLPM_genr_m2_dti <- data.frame()

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#Run models
#Model 1
#T1w
for(x in structs3_genr[1:43]){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
```

```
  ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
  sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

  #Estimate time independent predictors
  ' \n \n sum_dp_5_sqrt_scaled ~ sex \n ',
  x,"_f05_scaled", ' ~ sex \n

  #Estimate time dependent predictors
  sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
  sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
  sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
  x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 \n ',
  x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
  x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
  x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
CLPM_genr_fit <- lavaan(CLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                               survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_CLPM_genr_m1_t1w[(rowcount_clpm+2),1] <- x
#Which time point
results_CLPM_genr_m1_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <- c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_CLPM_genr_m1_t1w[c((rowcount_clpm+2)),3:5] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(20),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_genr_m1_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
#MRI
results_CLPM_genr_m1_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
```

```
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_genr_m1_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_m1_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_m1_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_m1_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_m1_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_m1_t1w[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_genr_m1_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m1_t1w[,3])-
                 mean(as.numeric(rsquared_CLPM_genr_m1_t1w[,3])))
  rsquared_CLPM_genr_m1_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m1_t1w[,5])-
                 mean(as.numeric(rsquared_CLPM_genr_m1_t1w[,5])))
  rsquared_CLPM_genr_m1_t1w[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m1_t1w[,7])-
                 mean(as.numeric(rsquared_CLPM_genr_m1_t1w[,7])))

#Reset rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#DTI
for(x in structs3_genr[44:57]){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",
```

```
  #Estimate time independent predictors
  ' \n \n sum_dp_5_sqrt_scaled ~ sex \n ',
  x,"_f05_scaled", ' ~ sex \n

  #Estimate time dependent predictors
  sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
  sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
  sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
  x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 \n ',
  x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
  x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
  x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

#Fit the model specified above
CLPM_genr_fit <- lavaan(CLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                               survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_CLPM_genr_m1_dti[(rowcount_clpm+2),1] <- x
#Which time point
results_CLPM_genr_m1_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <- c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_CLPM_genr_m1_dti[c((rowcount_clpm+2)),3:5] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(20),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_genr_m1_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
#MRI
results_CLPM_genr_m1_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
```

```
  results_CLPM_genr_m1_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_m1_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_m1_dti[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_m1_dti[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_m1_dti[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_m1_dti[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_genr_m1_dti[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m1_dti[,3])-
                 mean(as.numeric(rsquared_CLPM_genr_m1_dti[,3])))
  rsquared_CLPM_genr_m1_dti[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m1_dti[,5])-
                 mean(as.numeric(rsquared_CLPM_genr_m1_dti[,5])))
  rsquared_CLPM_genr_m1_dti[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m1_dti[,7])-
                 mean(as.numeric(rsquared_CLPM_genr_m1_dti[,7])))

#Reset rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#Model 2
#T1w
for(x in structs3_genr[1:43]){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_5_sqrt_scaled ~ sex +  maternal_education_middle +
```

```
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
  x,"_f05_scaled", ' ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

  #Estimate time dependent predictors
  sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
  sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
  sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
  x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 \n ',
  x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
  x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
  x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
CLPM_genr_fit <- lavaan(CLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                               survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_CLPM_genr_m2_t1w[(rowcount_clpm+2),1] <- x
#Which time point
results_CLPM_genr_m2_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <- c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_CLPM_genr_m2_t1w[c((rowcount_clpm+2)),3:5] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(28),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_genr_m2_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
#MRI
results_CLPM_genr_m2_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
```

```
  results_CLPM_genr_m2_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_m2_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_m2_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_m2_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_m2_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_m2_t1w[rowcount_fit, 1:7] <- c(x, lavInspect(CLPM_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_genr_m2_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m2_t1w[,3])-
                 mean(as.numeric(rsquared_CLPM_genr_m2_t1w[,3])))
  rsquared_CLPM_genr_m2_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m2_t1w[,5])-
                 mean(as.numeric(rsquared_CLPM_genr_m2_t1w[,5])))
  rsquared_CLPM_genr_m2_t1w[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m2_t1w[,7])-
                 mean(as.numeric(rsquared_CLPM_genr_m2_t1w[,7])))

#Reset rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#DTI
for(x in structs3_genr[44:57]){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_5_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
    x,"_f05_scaled", ' ~ sex + maternal_education_middle +
```

```
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

  #Estimate time dependent predictors
  sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
  sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
  sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
  x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 \n ',
  x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
  x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
  x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

#Fit the model specified above
CLPM_genr_fit <- lavaan(CLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                               survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_CLPM_genr_m2_dti[(rowcount_clpm+2),1] <- x
#Which time point
results_CLPM_genr_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <- c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_CLPM_genr_m2_dti[c((rowcount_clpm+2)),3:5] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(28),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_genr_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
#MRI
results_CLPM_genr_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_CLPM_genr_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
```

```
  #MRI -> CBCL
  results_CLPM_genr_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_m2_dti[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_m2_dti[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_m2_dti[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_m2_dti[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_genr_m2_dti[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m2_dti[,3])-
                 mean(as.numeric(rsquared_CLPM_genr_m2_dti[,3])))
  rsquared_CLPM_genr_m2_dti[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m2_dti[,5])-
                 mean(as.numeric(rsquared_CLPM_genr_m2_dti[,5])))
  rsquared_CLPM_genr_m2_dti[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m2_dti[,7])-
                 mean(as.numeric(rsquared_CLPM_genr_m2_dti[,7])))

#Reset rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#Model 3
#Specify the model for all brain morphology measures
for(x in structs3_genr[1:43]){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_5_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch +
    child_nationalorigin_wes \n ',
```

```
  x,"_f05_scaled", ' ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch +
  child_nationalorigin_wes \n

  #Estimate time dependent predictors
  sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
  sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
  sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
  x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled \n ',
  x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled \n ',
  x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
  x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
CLPM_genr_fit <- lavaan(CLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                               survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_CLPM_genr_m3_t1w[(rowcount_clpm+2),1] <- x
#Which time point
results_CLPM_genr_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <- c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_CLPM_genr_m3_t1w[c((rowcount_clpm+2)),3:5] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(31),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_genr_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
#MRI
results_CLPM_genr_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
  summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
```

```
  results_CLPM_genr_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_m3_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_m3_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_m3_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_m3_t1w[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #First calculate mean Rsquared
  mean_rsquared_CLPM_genr_m3_t1w <-
    mean(as.numeric(c(rsquared_CLPM_genr_m3_t1w[,3],
                      rsquared_CLPM_genr_m3_t1w[,5],
                      rsquared_CLPM_genr_m3_t1w[,7])))
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_genr_m3_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m3_t1w[,3])-
                 mean(as.numeric(rsquared_CLPM_genr_m3_t1w[,3])))
  rsquared_CLPM_genr_m3_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m3_t1w[,5])-
                 mean(as.numeric(rsquared_CLPM_genr_m3_t1w[,5])))
  rsquared_CLPM_genr_m3_t1w[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_CLPM_genr_m3_t1w[,7])-
                 mean(as.numeric(rsquared_CLPM_genr_m3_t1w[,7])))

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_genr_m1_t1w, "results_CLPM_genr_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_genr_m2_t1w, "results_CLPM_genr_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_genr_m3_t1w, "results_CLPM_genr_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_m1_t1w, "fitmeaures_CLPM_genr_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_m2_t1w, "fitmeaures_CLPM_genr_m2_t1w.csv",
          row.names = F, quote = F)
```

```
write.csv(fitmeaures_CLPM_genr_m3_t1w, "fitmeaures_CLPM_genr_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_m1_t1w, "rsquared_CLPM_genr_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_m2_t1w, "rsquared_CLPM_genr_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_m3_t1w, "rsquared_CLPM_genr_m3_t1w.csv",
          row.names = F, quote = F)

write.csv(results_CLPM_genr_m1_dti, "results_CLPM_genr_m1_dti.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_genr_m2_dti, "results_CLPM_genr_m2_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_m1_dti, "fitmeaures_CLPM_genr_m1_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_m2_dti, "fitmeaures_CLPM_genr_m2_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_m1_dti, "rsquared_CLPM_genr_m1_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_m2_dti, "rsquared_CLPM_genr_m2_dti.csv",
          row.names = F, quote = F)
```

2.1.7. False Discovery Rate correction (Benjamini-Hochberg)

```
#Subset results into hypothesis driven and exploratory
results_CLPM_genr_m3_hypothesisdriven <-
  rbind(results_CLPM_genr_m3_t1w[c(37:38,33:34,81:88),],
        results_CLPM_genr_m2_dti[c(3:4,7:8,15:18,21:22,29:30),])

results_CLPM_genr_m3_exploratory <-
  rbind(results_CLPM_genr_m3_t1w[c(3:32,35:36,39:80),],
        results_CLPM_genr_m2_dti[c(5:6,9:14,19:20,23:28),])

fdr_hypothesisdriven_genr  <- matrix(nrow = nrow(results_CLPM_genr_m3_hypothesisdriven)*4.5, ncol = 3)
fdr_exploratory_genr  <- matrix(nrow = nrow(results_CLPM_genr_m3_exploratory)*4.5, ncol = 3)

#Bind all pvalues together
pvals_all_hypothesisdriven_genr <- c(results_CLPM_genr_m3_hypothesisdriven[,5],
                                     results_CLPM_genr_m3_hypothesisdriven[,8],
                                     results_CLPM_genr_m3_hypothesisdriven[,11],
                                     results_CLPM_genr_m3_hypothesisdriven[,14],
                                     results_CLPM_genr_m3_hypothesisdriven[,17])
pvals_all_exploratory_genr <- c(results_CLPM_genr_m3_exploratory[,5],
                                results_CLPM_genr_m3_exploratory[,8],
                                results_CLPM_genr_m3_exploratory[,11],
                                results_CLPM_genr_m3_exploratory[,14],
                                results_CLPM_genr_m3_exploratory[,17])

#Calculate threshold for each test
for(x in 1:nrow(fdr_hypothesisdriven_genr)){
  thresh_temp <- x/nrow(fdr_hypothesisdriven_genr)*0.05
  if(x == 1){
    thresholds_hypothesisdriven_genr <- thresh_temp
```

```
  } else {
    thresholds_hypothesisdriven_genr <- c(thresholds_hypothesisdriven_genr, thresh_temp)
  }
}

for(x in 1:nrow(fdr_exploratory_genr)){
  thresh_temp <- x/nrow(fdr_exploratory_genr)*0.05
  if(x == 1){
    thresholds_exploratory_genr <- thresh_temp
  } else {
    thresholds_exploratory_genr <- c(thresholds_exploratory_genr, thresh_temp)
  }
}

#Fill FDR dataframes
fdr_hypothesisdriven_genr[,1] <- sort(as.numeric(pvals_all_hypothesisdriven_genr))
fdr_hypothesisdriven_genr[,2] <- thresholds_hypothesisdriven_genr
fdr_hypothesisdriven_genr[,3] <-
  ifelse(as.numeric(fdr_hypothesisdriven_genr[,1]) <
           as.numeric(fdr_hypothesisdriven_genr[,2]), "sig", "nonsig")

fdr_exploratory_genr[,1] <- sort(as.numeric(pvals_all_exploratory_genr))
fdr_exploratory_genr[,2] <- thresholds_exploratory_genr
fdr_exploratory_genr[,3] <-
  ifelse(as.numeric(fdr_exploratory_genr[,1]) <
           as.numeric(fdr_exploratory_genr[,2]), "sig", "nonsig")

#Add correction to tables
#Store significance after FDR-BH as * in
results_CLPM_genr_m3_hypothesisdriven[,"sig_CS"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_hypothesisdriven[,5]) <= 0.0226851851851852, "*", "")
results_CLPM_genr_m3_hypothesisdriven[,"sig_AR_CBCL"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_hypothesisdriven[,8]) <= 0.0226851851851852, "*", "")
results_CLPM_genr_m3_hypothesisdriven[,"sig_AR_MRI"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_hypothesisdriven[,11]) <= 0.0226851851851852, "*", "")
results_CLPM_genr_m3_hypothesisdriven[,"sig_CL_CBCLMRI"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_hypothesisdriven[,14]) <= 0.0226851851851852, "*", "")
results_CLPM_genr_m3_hypothesisdriven[,"sig_CL_MRICBCL"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_hypothesisdriven[,17]) <= 0.0226851851851852, "*", "")

results_CLPM_genr_m3_exploratory[,"sig_CS"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_exploratory[,5]) <= 0.0225925925925926, "*", "")
results_CLPM_genr_m3_exploratory[,"sig_AR_CBCL"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_exploratory[,8]) <= 0.0225925925925926, "*", "")
results_CLPM_genr_m3_exploratory[,"sig_AR_MRI"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_exploratory[,11]) <= 0.0225925925925926, "*", "")
results_CLPM_genr_m3_exploratory[,"sig_CL_CBCLMRI"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_exploratory[,14]) <= 0.0225925925925926, "*", "")
results_CLPM_genr_m3_exploratory[,"sig_CL_MRICBCL"] <-
  ifelse(as.numeric(results_CLPM_genr_m3_exploratory[,17]) <= 0.0225925925925926, "*", "")

#Print tables
#Hypothesis driven
```

```
results_CLPM_genr_m3_hypothesisdriven[,c(3:17)] <-
  sapply(results_CLPM_genr_m3_hypothesisdriven[,c(3:17)], as.numeric)
results_CLPM_genr_m3_hypothesisdriven[,c(1:5,18,6:8,19,9:11,20,12:14,21,15:17,22)] %>%
  kable(digits = 2,
        format = "latex",
      caption="Results hypothesis driven CLPM Generation R",
      col.names = c("Brain region", "Timepoint",
                    "CS B", "SE", "p", "",
                    "AR CBCL B", "SE", "p", "",
                    "AR MRI B", "SE", "p", "",
                    "CL CBCL>MRI B", "SE", "p", "",
                    "CL MRI>CBCL B", "SE", "p", ""),
      align="r") %>%
  kable_classic(full_width = F, html_font = "helvetica")
```

Table 1: Results hypothesis driven CLPM Generation R

	Brain region	Timepoint	CS B	SE	p		AR CBCL B	SE	p		AR MRI B	SE	p		CL CBCL>MRI B	SE	p		CL MRI>CBCL B	SE	p
37	medialorbitofrontal_vol	T1-T2	0.02	0.06	0.77		0.65	0.06	0	*	0.68	0.07	0	*	-0.01	0.04	0.72		0.04	0.06	0.56
38	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.60	0.06	0	*	0.01	0.04	0.74		-0.03	0.05	0.53
33	lateralorbitofrontal_vol	T1-T2	-0.04	0.06	0.54		0.65	0.06	0	*	0.63	0.09	0	*	-0.05	0.04	0.15		0.03	0.07	0.67
34	NA	T2-T3	NA	NA	NA	NA	0.68	0.06	0	*	0.63	0.07	0	*	0.03	0.04	0.45		-0.08	0.06	0.15
81	Left_Amygdala_vol	T1-T2	-0.10	0.06	0.12		0.64	0.06	0	*	0.53	0.08	0	*	0.00	0.04	0.99		-0.04	0.06	0.53
82	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.81	0.06	0	*	-0.02	0.05	0.66		0.02	0.05	0.74
83	Right_Amygdala_vol	T1-T2	-0.06	0.06	0.33		0.65	0.06	0	*	0.53	0.07	0	*	0.00	0.04	1.00		-0.01	0.06	0.93
84	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.79	0.06	0	*	0.04	0.04	0.38		0.00	0.05	0.99
85	Left_Accumbens_area_vol	T1-T2	-0.12	0.08	0.16		0.66	0.06	0	*	0.59	0.08	0	*	0.07	0.05	0.20		0.04	0.06	0.51
86	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.61	0.06	0	*	0.03	0.06	0.57		-0.03	0.05	0.56
87	Right_Accumbens_area_vol	T1-T2	-0.08	0.08	0.36		0.65	0.06	0	*	0.65	0.06	0	*	0.05	0.05	0.25		0.03	0.07	0.61
88	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.66	0.06	0	*	-0.01	0.05	0.88		-0.04	0.05	0.50
3	cgc_FA	T1-T2	0.14	0.07	0.06		0.64	0.05	0	*	0.32	0.06	0	*	0.01	0.06	0.88		0.02	0.06	0.77
4	NA	T2-T3	NA	NA	NA	NA	0.70	0.05	0	*	0.50	0.10	0	*	0.02	0.05	0.70		0.00	0.05	0.94
7	unc_FA	T1-T2	-0.12	0.07	0.12		0.64	0.05	0	*	0.38	0.05	0	*	-0.04	0.05	0.49		-0.04	0.05	0.48
8	NA	T2-T3	NA	NA	NA	NA	0.70	0.06	0	*	0.66	0.11	0	*	-0.11	0.05	0.02	*	-0.05	0.06	0.39
15	fmi_FA	T1-T2	-0.08	0.07	0.24		0.65	0.05	0	*	0.38	0.06	0	*	0.00	0.04	0.95		0.01	0.06	0.90
16	NA	T2-T3	NA	NA	NA	NA	0.71	0.06	0	*	0.70	0.11	0	*	0.01	0.05	0.84		-0.09	0.06	0.11
17	cgc_MD	T1-T2	-0.04	0.06	0.56		0.65	0.05	0	*	0.48	0.07	0	*	0.08	0.06	0.16		0.06	0.06	0.29
18	NA	T2-T3	NA	NA	NA	NA	0.71	0.05	0	*	0.62	0.05	0	*	0.07	0.05	0.10		-0.06	0.06	0.28
21	unc_MD	T1-T2	-0.01	0.07	0.94		0.64	0.05	0	*	0.67	0.08	0	*	0.03	0.06	0.66		0.07	0.06	0.24
22	NA	T2-T3	NA	NA	NA	NA	0.71	0.06	0	*	0.78	0.04	0	*	0.01	0.05	0.82		-0.03	0.06	0.59
29	fmi_MD	T1-T2	0.13	0.06	0.04		0.64	0.05	0	*	0.25	0.08	0	*	0.00	0.05	0.92		0.08	0.05	0.13
30	NA	T2-T3	NA	NA	NA	NA	0.70	0.06	0	*	0.38	0.09	0	*	0.05	0.06	0.37		0.08	0.06	0.20

```
#Exploratory
results_CLPM_genr_m3_exploratory[,c(3:17)] <-
  sapply(results_CLPM_genr_m3_exploratory[,c(3:17)], as.numeric)
results_CLPM_genr_m3_exploratory[,c(1:5,18,6:8,19,9:11,20,12:14,21,15:17,22)] %>%
  kable(digits = 2,
        format = "latex",
      caption="Results exploratory CLPM Generation R",
      col.names = c("Brain region", "Timepoint",
                    "CS B", "SE", "p", "",
                    "AR CBCL B", "SE", "p", "",
                    "AR MRI B", "SE", "p", "",
                    "CL CBCL>MRI B", "SE", "p", "",
                    "CL MRI>CBCL B", "SE", "p", ""),
      align="r") %>%
  kable_classic(full_width = F, html_font = "helvetica")
```

Table 2: Results exploratory CLPM Generation R

	Brain region	Timepoint	CS B	SE	p		AR CBCL B	SE	p		AR MRI B	SE	p		CL CBCL>MRI B	SE	p		CL MRI>CBCL B	SE	p
3	Thalamus_Proper_vol	T1-T2	-0.06	0.06	0.27		0.65	0.06	0	*	0.64	0.06	0	*	-0.04	0.03	0.16		0.02	0.06	0.73
4	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.75	0.05	0	*	-0.04	0.03	0.22		-0.01	0.06	0.91
5	Caudate_vol	T1-T2	-0.08	0.07	0.19		0.65	0.06	0	*	0.96	0.03	0	*	-0.02	0.02	0.18		0.01	0.06	0.86
6	NA	T2-T3	NA	NA	NA	NA	0.68	0.06	0	*	1.02	0.02	0	*	-0.01	0.02	0.60		-0.10	0.06	0.08
7	Putamen_vol	T1-T2	-0.13	0.07	0.06		0.65	0.06	0	*	0.90	0.04	0	*	-0.03	0.03	0.24		0.00	0.07	0.98
8	NA	T2-T3	NA	NA	NA	NA	0.68	0.06	0	*	0.92	0.03	0	*	0.01	0.02	0.72		-0.08	0.06	0.22
9	Pallidum_vol	T1-T2	-0.05	0.06	0.45		0.65	0.06	0	*	0.67	0.07	0	*	-0.06	0.04	0.15		0.07	0.07	0.29
10	NA	T2-T3	NA	NA	NA	NA	0.68	0.06	0	*	0.70	0.05	0	*	-0.07	0.04	0.09		-0.06	0.04	0.20
11	Hippocampus_vol	T1-T2	-0.09	0.06	0.17		0.65	0.06	0	*	0.74	0.05	0	*	-0.07	0.03	0.02	*	0.01	0.06	0.80
12	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.86	0.03	0	*	-0.01	0.02	0.54		-0.03	0.05	0.51
13	bankssts_vol	T1-T2	-0.04	0.07	0.57		0.65	0.06	0	*	0.79	0.05	0	*	0.01	0.03	0.59		0.05	0.06	0.38
14	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.94	0.04	0	*	0.03	0.02	0.27		-0.04	0.06	0.45
15	caudalanteriorcingulate_vol	T1-T2	-0.07	0.06	0.22		0.65	0.06	0	*	0.86	0.05	0	*	0.02	0.02	0.36		0.02	0.07	0.75
16	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.96	0.03	0	*	-0.04	0.02	0.11		-0.04	0.06	0.49
17	caudalmiddlefrontal_vol	T1-T2	0.04	0.06	0.47		0.65	0.06	0	*	0.79	0.05	0	*	0.05	0.03	0.09		0.02	0.06	0.69
18	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.91	0.03	0	*	0.01	0.03	0.71		-0.07	0.05	0.17
19	cuneus_vol	T1-T2	-0.09	0.08	0.26		0.65	0.06	0	*	0.85	0.04	0	*	-0.02	0.02	0.41		0.03	0.06	0.64
20	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.93	0.03	0	*	0.01	0.02	0.69		0.01	0.06	0.85
21	entorhinal_vol	T1-T2	-0.06	0.07	0.39		0.65	0.06	0	*	0.65	0.07	0	*	-0.04	0.05	0.36		0.03	0.06	0.66
22	NA	T2-T3	NA	NA	NA	NA	0.70	0.06	0	*	0.63	0.06	0	*	-0.01	0.05	0.78		0.10	0.07	0.16
23	fusiform_vol	T1-T2	-0.13	0.06	0.04		0.66	0.06	0	*	0.71	0.07	0	*	-0.02	0.04	0.53		0.08	0.06	0.18
24	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.85	0.04	0	*	0.00	0.03	0.87		0.00	0.05	0.94
25	inferiorparietal_vol	T1-T2	-0.06	0.06	0.31		0.65	0.06	0	*	0.88	0.05	0	*	0.00	0.03	0.93		0.02	0.06	0.71
26	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.95	0.03	0	*	0.02	0.02	0.44		-0.02	0.05	0.62
27	inferiortemporal_vol	T1-T2	0.02	0.07	0.76		0.65	0.06	0	*	0.68	0.06	0	*	-0.02	0.04	0.56		-0.01	0.05	0.80
28	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.77	0.05	0	*	-0.02	0.03	0.60		-0.01	0.06	0.82
29	isthmuscingulate_vol	T1-T2	-0.06	0.06	0.31		0.65	0.06	0	*	0.89	0.04	0	*	0.00	0.03	0.85		0.03	0.06	0.64
30	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.95	0.03	0	*	0.01	0.02	0.58		-0.01	0.06	0.89
31	lateraloccipital_vol	T1-T2	-0.06	0.06	0.32		0.66	0.06	0	*	0.84	0.04	0	*	0.00	0.02	0.95		0.06	0.06	0.26
32	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.94	0.03	0	*	0.03	0.02	0.15		-0.02	0.05	0.72
35	lingual_vol	T1-T2	-0.10	0.08	0.21		0.66	0.06	0	*	0.90	0.05	0	*	0.01	0.02	0.57		0.06	0.06	0.33
36	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.92	0.02	0	*	0.02	0.02	0.43		-0.01	0.06	0.81
39	middletemporal_vol	T1-T2	0.04	0.07	0.57		0.65	0.06	0	*	0.54	0.07	0	*	-0.01	0.04	0.82		0.05	0.06	0.34
40	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.81	0.05	0	*	0.04	0.03	0.28		-0.05	0.05	0.39
41	parahippocampal_vol	T1-T2	-0.07	0.08	0.38		0.66	0.06	0	*	0.90	0.05	0	*	-0.02	0.04	0.53		0.08	0.05	0.10
42	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.85	0.03	0	*	0.01	0.03	0.72		0.01	0.05	0.81
43	paracentral_vol	T1-T2	-0.02	0.07	0.74		0.65	0.06	0	*	0.83	0.05	0	*	0.01	0.04	0.76		0.01	0.06	0.88
44	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.86	0.04	0	*	-0.02	0.03	0.46		0.04	0.05	0.49
45	parsopercularis_vol	T1-T2	-0.02	0.07	0.79		0.65	0.06	0	*	0.90	0.04	0	*	0.00	0.02	0.98		0.05	0.05	0.31
46	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.92	0.04	0	*	0.01	0.03	0.85		-0.04	0.05	0.43
47	parsorbitalis_vol	T1-T2	-0.06	0.06	0.34		0.65	0.06	0	*	0.64	0.07	0	*	-0.08	0.04	0.03		-0.03	0.06	0.68
48	NA	T2-T3	NA	NA	NA	NA	0.68	0.06	0	*	0.82	0.04	0	*	0.05	0.03	0.10		-0.07	0.06	0.22
49	parstriangularis_vol	T1-T2	-0.04	0.06	0.50		0.65	0.06	0	*	0.87	0.04	0	*	-0.04	0.03	0.15		0.02	0.07	0.76
50	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	1.01	0.03	0	*	0.04	0.02	0.06		-0.05	0.06	0.46
51	pericalcarine_vol	T1-T2	-0.08	0.08	0.26		0.66	0.06	0	*	0.93	0.04	0	*	-0.01	0.03	0.85		0.05	0.06	0.39
52	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.81	0.04	0	*	0.02	0.03	0.56		0.06	0.05	0.22
53	postcentral_vol	T1-T2	-0.02	0.06	0.75		0.65	0.06	0	*	0.82	0.07	0	*	-0.01	0.03	0.65		0.02	0.07	0.78
54	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.95	0.02	0	*	0.00	0.02	0.83		0.02	0.06	0.77
55	posteriorcingulate_vol	T1-T2	-0.04	0.06	0.50		0.66	0.06	0	*	0.96	0.05	0	*	0.01	0.02	0.65		0.07	0.06	0.24
56	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.93	0.03	0	*	0.01	0.02	0.80		0.05	0.05	0.25
57	precentral_vol	T1-T2	-0.04	0.06	0.54		0.65	0.06	0	*	0.75	0.07	0	*	0.01	0.03	0.79		-0.02	0.06	0.75
58	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.88	0.03	0	*	0.01	0.02	0.70		-0.04	0.06	0.53
59	precuneus_vol	T1-T2	-0.05	0.05	0.33		0.65	0.06	0	*	0.86	0.06	0	*	0.00	0.03	0.88		0.01	0.07	0.87
60	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.90	0.03	0	*	0.02	0.02	0.30		-0.07	0.06	0.24
61	rostralanteriorcingulate_vol	T1-T2	-0.04	0.06	0.52		0.65	0.06	0	*	0.84	0.06	0	*	0.00	0.02	0.99		0.02	0.06	0.75
62	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.94	0.04	0	*	-0.01	0.03	0.72		-0.07	0.05	0.18
63	rostralmiddlefrontal_vol	T1-T2	0.02	0.06	0.77		0.65	0.06	0	*	0.63	0.08	0	*	0.00	0.03	1.00		-0.01	0.06	0.83
64	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.88	0.04	0	*	0.04	0.03	0.16		-0.02	0.06	0.67
65	superiorfrontal_vol	T1-T2	-0.01	0.06	0.88		0.65	0.06	0	*	0.81	0.07	0	*	0.05	0.03	0.07		0.01	0.06	0.83
66	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.92	0.03	0	*	0.00	0.02	0.96		0.01	0.05	0.85
67	superiorparietal_vol	T1-T2	-0.02	0.06	0.78		0.65	0.06	0	*	0.79	0.06	0	*	0.02	0.03	0.60		-0.02	0.07	0.82
68	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.84	0.05	0	*	0.05	0.03	0.12		-0.03	0.05	0.53
69	superiortemporal_vol	T1-T2	0.02	0.06	0.74		0.65	0.06	0	*	0.75	0.08	0	*	-0.02	0.03	0.43		0.03	0.06	0.66
70	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.80	0.04	0	*	0.03	0.03	0.23		-0.06	0.06	0.29
71	supramarginal_vol	T1-T2	-0.02	0.05	0.71		0.65	0.06	0	*	0.89	0.05	0	*	-0.03	0.03	0.35		0.03	0.06	0.61
72	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.93	0.04	0	*	0.02	0.02	0.25		-0.01	0.06	0.81
73	frontalpole_vol	T1-T2	-0.05	0.08	0.51		0.65	0.06	0	*	0.41	0.07	0	*	-0.07	0.05	0.21		0.03	0.07	0.71
74	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.46	0.09	0	*	-0.04	0.07	0.60		-0.02	0.06	0.70
75	temporalpole_vol	T1-T2	-0.08	0.08	0.28		0.65	0.06	0	*	0.41	0.08	0	*	-0.19	0.07	0.01	*	-0.04	0.06	0.51
76	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.58	0.06	0	*	-0.01	0.06	0.85		0.01	0.05	0.87
77	transversetemporal_vol	T1-T2	-0.05	0.07	0.49		0.64	0.06	0	*	0.96	0.03	0	*	0.01	0.02	0.80		-0.05	0.06	0.40
78	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.92	0.03	0	*	0.02	0.02	0.53		0.02	0.07	0.80
79	insula_vol	T1-T2	-0.07	0.05	0.16		0.66	0.06	0	*	0.75	0.06	0	*	-0.02	0.03	0.56		0.06	0.06	0.38
80	NA	T2-T3	NA	NA	NA	NA	0.69	0.06	0	*	0.66	0.07	0	*	0.03	0.04	0.43		-0.05	0.05	0.40
510	cst_FA	T1-T2	-0.06	0.07	0.40		0.64	0.05	0	*	0.49	0.05	0	*	0.03	0.05	0.55		-0.01	0.06	0.83
610	NA	T2-T3	NA	NA	NA	NA	0.70	0.06	0	*	0.80	0.07	0	*	-0.12	0.05	0.01	*	-0.01	0.05	0.84
91	ilf_FA	T1-T2	-0.01	0.08	0.93		0.64	0.05	0	*	0.43	0.06	0	*	-0.05	0.05	0.27		-0.01	0.06	0.80
101	NA	T2-T3	NA	NA	NA	NA	0.70	0.06	0	*	0.73	0.10	0	*	-0.09	0.05	0.09		-0.04	0.05	0.48
111	slf_FA	T1-T2	0.03	0.07	0.69		0.65	0.05	0	*	0.40	0.05	0	*	-0.05	0.05	0.26		-0.03	0.06	0.60
121	NA	T2-T3	NA	NA	NA	NA	0.70	0.06	0	*	0.77	0.11	0	*	-0.05	0.05	0.31		-0.02	0.05	0.65
131	fma_FA	T1-T2	0.00	0.07	0.99		0.64	0.05	0	*	0.28	0.06	0	*	-0.04	0.05	0.43		-0.08	0.05	0.15
141	NA	T2-T3	NA	NA	NA	NA	0.70	0.06	0	*	0.49	0.13	0	*	-0.13	0.06	0.04		-0.05	0.05	0.40
191	cst_MD	T1-T2	-0.13	0.07	0.07		0.64	0.05	0	*	0.27	0.07	0	*	-0.06	0.06	0.31		-0.02	0.06	0.80
201	NA	T2-T3	NA	NA	NA	NA	0.70	0.06	0	*	0.54	0.06	0	*	0.07	0.06	0.21		-0.09	0.06	0.17
231	ilf_MD	T1-T2	0.00	0.07	0.99		0.65	0.05	0	*	0.72	0.07	0	*	0.03	0.04	0.45		0.08	0.05	0.12
241	NA	T2-T3	NA	NA	NA	NA	0.70	0.06	0	*	0.80	0.05	0	*	-0.01	0.04	0.78		0.01	0.05	0.89
251	slf_MD	T1-T2	-0.06	0.07	0.35		0.65	0.05	0	*	0.78	0.04	0	*	0.01	0.04	0.88		0.07	0.06	0.24
261	NA	T2-T3	NA	NA	NA	NA	0.70	0.05	0	*	0.87	0.04	0	*	0.00	0.03	0.91		-0.04	0.06	0.48
271	fma_MD	T1-T2	0.03	0.08	0.69		0.64	0.05	0	*	0.48	0.08	0	*	-0.01	0.05	0.84		0.07	0.05	0.17
281	NA	T2-T3	NA	NA	NA	NA	0.71	0.05	0	*	0.63	0.13	0	*	0.11	0.06	0.04		-0.07	0.05	0.11

```
#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_genr_m3_hypothesisdriven, "results_CLPM_genr_m3_hypothesisdriven.csv",
```

```
          row.names = F, quote = F)
write.csv(results_CLPM_genr_m3_exploratory, "results_CLPM_genr_m3_exploratory.csv",
          row.names = F, quote = F)
```

2.1.8. Random intercept cross-lagged panel model (RI-CLPM)

For the model set-up of the RI-CLPM, we have used the syntax from Jeroen Mulder as an example. This syntax is available via: https://jeroendmulder.github.io/RI-CLPM/lavaan.html

```
#Create empty dataframes to store results
results_RICLPM_genr_m1_t1w <- data.frame()
fitmeaures_RICLPM_genr_m1_t1w <- data.frame()
rsquared_RICLPM_genr_m1_t1w <- data.frame()
results_RICLPM_genr_m2_t1w <- data.frame()
fitmeaures_RICLPM_genr_m2_t1w <- data.frame()
rsquared_RICLPM_genr_m2_t1w <- data.frame()
results_RICLPM_genr_m3_t1w <- data.frame()
fitmeaures_RICLPM_genr_m3_t1w <- data.frame()
rsquared_RICLPM_genr_m3_t1w <- data.frame()

results_RICLPM_genr_m1_dti <- data.frame()
fitmeaures_RICLPM_genr_m1_dti <- data.frame()
rsquared_RICLPM_genr_m1_dti <- data.frame()
results_RICLPM_genr_m2_dti <- data.frame()
fitmeaures_RICLPM_genr_m2_dti <- data.frame()
rsquared_RICLPM_genr_m2_dti <- data.frame()

#Model 1
#T1w
#Specify rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
#Caudate (2), hippocampus (5) and pars triangularis (24) are skipped
#as these models do not converge
for(x in structs3_genr[c(1,3:4,6:23,25:43)]){
  RICLPM_genr <- paste0('
                        # Create between components (random intercepts)
                        RIx =~ 1*sum_dp_5_sqrt_scaled + 1*sum_dp_9m_sqrt_scaled +
                        1*sum_dp_14_sqrt_scaled
                        RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled",
'                        \n + 1*', x,"_f13_scaled",

                        # Create within-person centered variables
                        ' \n  wx1 =~ 1*sum_dp_5_sqrt_scaled
                        wx2 =~ 1*sum_dp_9m_sqrt_scaled
                        wx3 =~ 1*sum_dp_14_sqrt_scaled
                        wy1 =~ 1* \n', x,"_f05_scaled",
                        ' \n wy2 =~ 1* \n', x,"_f09_scaled",
                        ' \n wy3 =~ 1* \n', x,"_f13_scaled",

                        # Estimate the lagged effects between the variables
                        ' \n wx2 + wy2 ~ wx1 + wy1
```

```
                        wx3 + wy3 ~ wx2 + wy2

                        #Estimate time independent predictors
                        RIx ~ sex
                        RIy ~ sex

                        #Estimate time dependent predictors
                        sum_dp_5_sqrt_scaled ~ agechild_GR1075
                        sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m
                        sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093',
                        ' \n ', x,"_f05_scaled", '~ age_child_mri_f05 + HC12_F5',
                        ' \n ', x,"_f09_scaled", '~ age_child_mri_f09 + HC12_F9',
                        ' \n ', x,"_f13_scaled", '~ age_child_mri_f13 + HC12_F13',

                        # Estimate the covariance between variables at the first wave.
                        ' \n wx1 ~~ wy1 # Covariance

                        # Estimate the covariances between the residuals of variables
                        wx2 ~~ wy2
                        wx3 ~~ wy3

                        # Estimate the variance and covariance of the random intercepts.
                        RIx ~~ RIx
                        RIy ~~ RIy
                        RIx ~~ RIy

                        # Estimate the (residual) variance of the within-person centered variables.
                        wx1 ~~ wx1 # Variances
                        wy1 ~~ wy1
                        wx2 ~~ wx2 # Residual variances
                        wy2 ~~ wy2
                        wx3 ~~ wx3
                        wy3 ~~ wy3
                        ')

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
RICLPM_genr_fit <- lavaan(RICLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                 survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_RICLPM_genr_m1_t1w[(rowcount_riclpm+2),1] <- x
#Which time point
results_RICLPM_genr_m1_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),2] <-
  c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_RICLPM_genr_m1_t1w[c((rowcount_riclpm+2)),3:5] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(32),c(5,6,8)]
#Autoregressive parameters
```

```
  #CBCL
  results_RICLPM_genr_m1_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),6:8] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(13,17),c(5,6,8)]
  #MRI
  results_RICLPM_genr_m1_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),9:11] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(16,20),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_RICLPM_genr_m1_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),12:14] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(15,19),c(5,6,8)]
  #MRI -> CBCL
  results_RICLPM_genr_m1_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),15:17] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14,18),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_RICLPM_genr_m1_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_RICLPM_genr_m1_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                                      "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                      "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_RICLPM_genr_m1_t1w[rowcount_fit, 1:5] <-
    c(x, summary(RICLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_RICLPM_genr_m1_t1w[rowcount_fit, 1:9] <-
    c(x, lavInspect(RICLPM_genr_CSA, "rsquare")[1:8])
  #Adapt rowcounts to make sure results are stored properly
  rowcount_riclpm <- rowcount_riclpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_RICLPM_genr_m1_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m1_t1w[,5])-
                 mean(as.numeric(rsquared_RICLPM_genr_m1_t1w[,5])))
  rsquared_RICLPM_genr_m1_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m1_t1w[,7])-
                 mean(as.numeric(rsquared_RICLPM_genr_m1_t1w[,7])))
  rsquared_RICLPM_genr_m1_t1w[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m1_t1w[,9])-
                 mean(as.numeric(rsquared_RICLPM_genr_m1_t1w[,9])))

#DTI
#Specify rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
for(x in structs3_genr[c(44:57)]){
  RICLPM_genr <- paste0('
                        # Create between components (random intercepts)
                        RIx =~ 1*sum_dp_5_sqrt_scaled + 1*sum_dp_9m_sqrt_scaled
```

```
                        + 1*sum_dp_14_sqrt_scaled
                        RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled", ' \n
                        + 1*', x,"_f13_scaled",

                        # Create within-person centered variables
                        ' \n  wx1 =~ 1*sum_dp_5_sqrt_scaled
                        wx2 =~ 1*sum_dp_9m_sqrt_scaled
                        wx3 =~ 1*sum_dp_14_sqrt_scaled
                        wy1 =~ 1* \n', x,"_f05_scaled",
                        ' \n wy2 =~ 1* \n', x,"_f09_scaled",
                        ' \n wy3 =~ 1* \n', x,"_f13_scaled",

                        # Estimate the lagged effects between the variables
                        ' \n wx2 + wy2 ~ wx1 + wy1
                        wx3 + wy3 ~ wx2 + wy2

                        #Estimate time independent predictors
                        RIx ~ sex
                        RIy ~ sex

                        #Estimate time dependent predictors
                        sum_dp_5_sqrt_scaled ~ agechild_GR1075
                        sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m
                        sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093',
                        ' \n ', x,"_f05_scaled", '~ age_child_mri_f05 + HC12_F5',
                        ' \n ', x,"_f09_scaled", '~ age_child_mri_f09 + HC12_F9',
                        ' \n ', x,"_f13_scaled", '~ age_child_mri_f13 + HC12_F13',

                        # Estimate the covariance between variables at the first wave.
                        ' \n wx1 ~~ wy1 # Covariance

                        # Estimate the covariances between the residuals of variables
                        wx2 ~~ wy2
                        wx3 ~~ wy3

                        # Estimate the variance and covariance of the random intercepts.
                        RIx ~~ RIx
                        RIy ~~ RIy
                        RIx ~~ RIy

                        # Estimate the (residual) variance of the within-person centered variables.
                        wx1 ~~ wx1 # Variances
                        wy1 ~~ wy1
                        wx2 ~~ wx2 # Residual variances
                        wy2 ~~ wy2
                        wx3 ~~ wx3
                        wy3 ~~ wy3
                        ')

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

#Fit the model specified above
```

```
  RICLPM_genr_fit <- lavaan(RICLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                   survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_RICLPM_genr_m1_dti[(rowcount_riclpm+2),1] <- x
  #Which time point
  results_RICLPM_genr_m1_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),2] <- c("T1-T2", "T2-T3")
  #Cross-sectional associations T1
  results_RICLPM_genr_m1_dti[c((rowcount_riclpm+2)),3:5] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(32),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_RICLPM_genr_m1_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),6:8] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(13,17),c(5,6,8)]
  #MRI
  results_RICLPM_genr_m1_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),9:11] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(16,20),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_RICLPM_genr_m1_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),12:14] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(15,19),c(5,6,8)]
  #MRI -> CBCL
  results_RICLPM_genr_m1_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),15:17] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14,18),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_RICLPM_genr_m1_dti[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_RICLPM_genr_m1_dti[2,] <-
    c("Brain region", "Wave", "B", "S.E.", "p-value",
      "B", "S.E.", "p-value", "B", "S.E.", "p-value",
      "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_RICLPM_genr_m1_dti[rowcount_fit, 1:5] <-
    c(x, summary(RICLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_RICLPM_genr_m1_dti[rowcount_fit, 1:9] <-
    c(x, lavInspect(RICLPM_genr_CSA, "rsquare")[1:8])
  #Adapt rowcounts to make sure results are stored properly
  rowcount_riclpm <- rowcount_riclpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_RICLPM_genr_m1_dti[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m1_dti[,5])-
                 mean(as.numeric(rsquared_RICLPM_genr_m1_dti[,5])))
  rsquared_RICLPM_genr_m1_dti[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m1_dti[,7])-
                 mean(as.numeric(rsquared_RICLPM_genr_m1_dti[,7])))
```

```
  rsquared_RICLPM_genr_m1_dti[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m1_dti[,9])-
                 mean(as.numeric(rsquared_RICLPM_genr_m1_dti[,9])))

#Model 2
#T1w
#Reset rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
#Caudate (2), isthmus cingulate (14) and
#pars triangularis (24) do not converge
for(x in structs3_genr[c(1,3:13,15:23,25:43)]){
  RICLPM_genr <- paste0('
                        # Create between components (random intercepts)
                        RIx =~ 1*sum_dp_5_sqrt_scaled + 1*sum_dp_9m_sqrt_scaled
                        + 1*sum_dp_14_sqrt_scaled
                        RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled", ' \n
                        + 1*', x,"_f13_scaled",

                        # Create within-person centered variables
                        ' \n  wx1 =~ 1*sum_dp_5_sqrt_scaled
                        wx2 =~ 1*sum_dp_9m_sqrt_scaled
                        wx3 =~ 1*sum_dp_14_sqrt_scaled
                        wy1 =~ 1* \n', x,"_f05_scaled",
                        ' \n wy2 =~ 1* \n', x,"_f09_scaled",
                        ' \n wy3 =~ 1* \n', x,"_f13_scaled",

                        # Estimate the lagged effects between the variables
                        ' \n wx2 + wy2 ~ wx1 + wy1
                        wx3 + wy3 ~ wx2 + wy2

                        #Estimate time independent predictors
                        RIx ~ sex + maternal_education_middle +
                        maternal_education_high + child_nationalorigin_dutch +
                        child_nationalorigin_wes
                        RIy ~ sex + maternal_education_middle +
                        maternal_education_high + child_nationalorigin_dutch +
                        child_nationalorigin_wes

                        #Estimate time dependent predictors
                        wx1 ~ agechild_GR1075
                        wx2 ~ AgeChild_CBCL9m
                        wx3 ~ AGECHILD_GR1093
                        wy1 ~ age_child_mri_f05 + HC12_F5
                        wy2 ~ age_child_mri_f09 + HC12_F9
                        wy3 ~ age_child_mri_f13 + HC12_F13

                        # Estimate the covariance between variables at the first wave.
                        wx1 ~~ wy1 # Covariance

                        # Estimate the covariances between the residuals of variables
```

```
                        wx2 ~~ wy2
                        wx3 ~~ wy3

                        # Estimate the variance and covariance of the random intercepts.
                        RIx ~~ RIx
                        RIy ~~ RIy
                        RIx ~~ RIy

                        # Estimate the (residual) variance of the within-person centered variables.
                        wx1 ~~ wx1 # Variances
                        wy1 ~~ wy1
                        wx2 ~~ wx2 # Residual variances
                        wy2 ~~ wy2
                        wx3 ~~ wx3
                        wy3 ~~ wy3
                        ')

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
RICLPM_genr_fit <- lavaan(RICLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                 survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_RICLPM_genr_m2_t1w[(rowcount_riclpm+2),1] <- x
#Which time point
results_RICLPM_genr_m2_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),2] <-
  c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_RICLPM_genr_m2_t1w[c((rowcount_riclpm+2)),3:5] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(40),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_RICLPM_genr_m2_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),6:8] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(13,17),c(5,6,8)]
#MRI
results_RICLPM_genr_m2_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),9:11] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(16,20),c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_RICLPM_genr_m2_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),12:14] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(15,19),c(5,6,8)]
#MRI -> CBCL
results_RICLPM_genr_m2_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),15:17] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14,18),c(5,6,8)]
#Which measure (actually only needs to be specified once)
results_RICLPM_genr_m2_t1w[1,c(4,7,10,13,16)] <-
  c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
results_RICLPM_genr_m2_t1w[2,] <-
  c("Brain region", "Wave", "B", "S.E.", "p-value",
```

```
      "B", "S.E.", "p-value", "B", "S.E.", "p-value",
      "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_RICLPM_genr_m2_t1w[rowcount_fit, 1:4] <-
    summary(RICLPM_genr_CSA, fit.measures = T,
            standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")]
  #Store raw Rsquared
  rsquared_RICLPM_genr_m2_t1w[rowcount_fit, 1:9] <-
    c(x, lavInspect(RICLPM_genr_CSA, "rsquare")[1:8])
  #Adapt rowcounts to make sure results are stored properly
  rowcount_riclpm <- rowcount_riclpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_RICLPM_genr_m2_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m2_t1w[,5])-
                 mean(as.numeric(rsquared_RICLPM_genr_m2_t1w[,5])))
  rsquared_RICLPM_genr_m2_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m2_t1w[,7])-
                 mean(as.numeric(rsquared_RICLPM_genr_m2_t1w[,7])))
  rsquared_RICLPM_genr_m2_t1w[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m2_t1w[,9])-
                 mean(as.numeric(rsquared_RICLPM_genr_m2_t1w[,9])))


#Reset rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#DTI
#Specify the model for all brain morphology measures
for(x in structs3_genr[c(44:57)]){
  RICLPM_genr <- paste0('
                        # Create between components (random intercepts)
                        RIx =~ 1*sum_dp_5_sqrt_scaled + 1*sum_dp_9m_sqrt_scaled +
                        1*sum_dp_14_sqrt_scaled
                        RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled", ' \n +
                        1*', x,"_f13_scaled",

                        # Create within-person centered variables
                        ' \n  wx1 =~ 1*sum_dp_5_sqrt_scaled
                        wx2 =~ 1*sum_dp_9m_sqrt_scaled
                        wx3 =~ 1*sum_dp_14_sqrt_scaled
                        wy1 =~ 1* \n', x,"_f05_scaled",
                        ' \n wy2 =~ 1* \n', x,"_f09_scaled",
                        ' \n wy3 =~ 1* \n', x,"_f13_scaled",

                        # Estimate the lagged effects between the variables
                        ' \n wx2 + wy2 ~ wx1 + wy1
                        wx3 + wy3 ~ wx2 + wy2
```

```
                        #Estimate time independent predictors
                        RIx ~ sex + maternal_education_middle +
                        maternal_education_high + child_nationalorigin_dutch +
                        child_nationalorigin_wes
                        RIy ~ sex + maternal_education_middle +
                        maternal_education_high + child_nationalorigin_dutch +
                        child_nationalorigin_wes

                        #Estimate time dependent predictors
                        wx1 ~ agechild_GR1075
                        wx2 ~ AgeChild_CBCL9m
                        wx3 ~ AGECHILD_GR1093
                        wy1 ~ age_child_mri_f05 + HC12_F5
                        wy2 ~ age_child_mri_f09 + HC12_F9
                        wy3 ~ age_child_mri_f13 + HC12_F13

                        # Estimate the covariance between variables at the first wave.
                        wx1 ~~ wy1 # Covariance

                        # Estimate the covariances between the residuals of variables
                        wx2 ~~ wy2
                        wx3 ~~ wy3

                        # Estimate the variance and covariance of the random intercepts.
                        RIx ~~ RIx
                        RIy ~~ RIy
                        RIx ~~ RIy

                        # Estimate the (residual) variance of the within-person centered variables.
                        wx1 ~~ wx1 # Variances
                        wy1 ~~ wy1
                        wx2 ~~ wx2 # Residual variances
                        wy2 ~~ wy2
                        wx3 ~~ wx3
                        wy3 ~~ wy3
                        ')

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

#Fit the model specified above
RICLPM_genr_fit <- lavaan(RICLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                 survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_RICLPM_genr_m2_dti[(rowcount_riclpm+2),1] <- x
#Which time point
results_RICLPM_genr_m2_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),2] <-
  c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_RICLPM_genr_m2_dti[c((rowcount_riclpm+2)),3:5] <-
```

```
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(40),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_RICLPM_genr_m2_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),6:8] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(13,17),c(5,6,8)]
  #MRI
  results_RICLPM_genr_m2_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),9:11] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(16,20),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_RICLPM_genr_m2_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),12:14] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(15,19),c(5,6,8)]
  #MRI -> CBCL
  results_RICLPM_genr_m2_dti[c((rowcount_riclpm+2):(rowcount_riclpm+3)),15:17] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14,18),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_RICLPM_genr_m2_dti[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_RICLPM_genr_m2_dti[2,] <-
    c("Brain region", "Wave", "B", "S.E.", "p-value",
      "B", "S.E.", "p-value", "B", "S.E.", "p-value",
      "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_RICLPM_genr_m2_dti[rowcount_fit, 1:4] <-
    summary(RICLPM_genr_CSA, fit.measures = T,
            standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")]
  #Store raw Rsquared
  rsquared_RICLPM_genr_m2_dti[rowcount_fit, 1:9] <-
    c(x, lavInspect(RICLPM_genr_CSA, "rsquare")[1:8])
  #Adapt rowcounts to make sure results are stored properly
  rowcount_riclpm <- rowcount_riclpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_RICLPM_genr_m2_dti[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m2_dti[,5])-
                 mean(as.numeric(rsquared_RICLPM_genr_m2_dti[,5])))
  rsquared_RICLPM_genr_m2_dti[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m2_dti[,7])-
                 mean(as.numeric(rsquared_RICLPM_genr_m2_dti[,7])))
  rsquared_RICLPM_genr_m2_dti[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m2_dti[,9])-
                 mean(as.numeric(rsquared_RICLPM_genr_m2_dti[,9])))

#Reset rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#Model 3
#T1w
```

```
#Reset rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
for(x in structs3_genr[1:43]){
  RICLPM_genr <- paste0('
  # Create between components (random intercepts)
  RIx =~ 1*sum_dp_5_sqrt_scaled + 1*sum_dp_9m_sqrt_scaled + 1*sum_dp_14_sqrt_scaled
  RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled", ' \n + 1*', x,"_f13_scaled",

  # Create within-person centered variables
  ' \n  wx1 =~ 1*sum_dp_5_sqrt_scaled
  wx2 =~ 1*sum_dp_9m_sqrt_scaled
  wx3 =~ 1*sum_dp_14_sqrt_scaled
  wy1 =~ 1* \n', x,"_f05_scaled",
  ' \n wy2 =~ 1* \n', x,"_f09_scaled",
  ' \n wy3 =~ 1* \n', x,"_f13_scaled",

  # Estimate the lagged effects between the variables
  ' \n wx2 + wy2 ~ wx1 + wy1
  wx3 + wy3 ~ wx2 + wy2

  #Estimate time independent predictors
  RIx ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes
  RIy ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes

  #Estimate time dependent predictors
  wx1 ~ agechild_GR1075
  wx2 ~ AgeChild_CBCL9m
  wx3 ~ AGECHILD_GR1093
  wy1 ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled
  wy2 ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled
  wy3 ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled

  # Estimate the covariance between variables at the first wave.
  wx1 ~~ wy1 # Covariance

  # Estimate the covariances between the residuals of variables
  wx2 ~~ wy2
  wx3 ~~ wy3

  # Estimate the variance and covariance of the random intercepts.
  RIx ~~ RIx
  RIy ~~ RIy
  RIx ~~ RIy

  # Estimate the (residual) variance of the within-person centered variables.
  wx1 ~~ wx1 # Variances
  wy1 ~~ wy1
  wx2 ~~ wx2 # Residual variances
```

```
wy2 ~~ wy2
wx3 ~~ wx3
wy3 ~~ wy3
')

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
RICLPM_genr_fit <- lavaan(RICLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                 survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_RICLPM_genr_m3_t1w[(rowcount_riclpm+2),1] <- x
#Which time point
results_RICLPM_genr_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),2] <-
  c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_RICLPM_genr_m3_t1w[c((rowcount_riclpm+2)),3:5] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(43),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_RICLPM_genr_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),6:8] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(13,17),c(5,6,8)]
#MRI
results_RICLPM_genr_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),9:11] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(16,20),c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_RICLPM_genr_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),12:14] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(15,19),c(5,6,8)]
#MRI -> CBCL
results_RICLPM_genr_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),15:17] <-
  summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14,18),c(5,6,8)]
#Which measure (actually only needs to be specified once)
results_RICLPM_genr_m3_t1w[1,c(4,7,10,13,16)] <-
  c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
results_RICLPM_genr_m3_t1w[2,] <-
  c("Brain region", "Wave", "B", "S.E.", "p-value",
    "B", "S.E.", "p-value", "B", "S.E.", "p-value",
    "B", "S.E.", "p-value", "B", "S.E.", "p-value")
#Store summary statistics
fitmeaures_RICLPM_genr_m3_t1w[rowcount_fit, 1:4] <-
  summary(RICLPM_genr_CSA, fit.measures = T,
          standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")]
#Store raw Rsquared
rsquared_RICLPM_genr_m3_t1w[rowcount_fit, 1:9] <-
  c(x, lavInspect(RICLPM_genr_CSA, "rsquare")[1:8])
#Adapt rowcounts to make sure results are stored properly
rowcount_riclpm <- rowcount_riclpm + 2
rowcount_fit <- rowcount_fit + 1
```

```
}
#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_RICLPM_genr_m3_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m3_t1w[,5])-
                 mean(as.numeric(rsquared_RICLPM_genr_m3_t1w[,5]), na.rm = T))
  rsquared_RICLPM_genr_m3_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m3_t1w[,7])-
                 mean(as.numeric(rsquared_RICLPM_genr_m3_t1w[,7]), na.rm = T))
  rsquared_RICLPM_genr_m3_t1w[,"relative_MRI_T3"] <-
    as.numeric(as.numeric(rsquared_RICLPM_genr_m3_t1w[,9])-
                 mean(as.numeric(rsquared_RICLPM_genr_m3_t1w[,9]), na.rm = T))

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_RICLPM_genr_m1_t1w, "results_RICLPM_genr_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(results_RICLPM_genr_m2_t1w, "results_RICLPM_genr_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(results_RICLPM_genr_m3_t1w, "results_RICLPM_genr_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_RICLPM_genr_m1_t1w, "fitmeaures_RICLPM_genr_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_RICLPM_genr_m2_t1w, "fitmeaures_RICLPM_genr_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_RICLPM_genr_m3_t1w, "fitmeaures_RICLPM_genr_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_RICLPM_genr_m1_t1w, "rsquared_RICLPM_genr_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_RICLPM_genr_m2_t1w, "rsquared_RICLPM_genr_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_RICLPM_genr_m3_t1w, "rsquared_RICLPM_genr_m3_t1w.csv",
          row.names = F, quote = F)

write.csv(results_RICLPM_genr_m1_dti, "results_RICLPM_genr_m1_dti.csv",
          row.names = F, quote = F)
write.csv(results_RICLPM_genr_m2_dti, "results_RICLPM_genr_m2_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_RICLPM_genr_m1_dti, "fitmeaures_RICLPM_genr_m1_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_RICLPM_genr_m2_dti, "fitmeaures_RICLPM_genr_m2_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_RICLPM_genr_m1_dti, "rsquared_RICLPM_genr_m1_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_RICLPM_genr_m2_dti, "rsquared_RICLPM_genr_m2_dti.csv",
          row.names = F, quote = F)
```

2.1.9. False Discovery Rate correction (Benjamini-Hochberg)

```
#Subset results into hypothesis driven and exploratory
results_RICLPM_genr_m3_hypothesisdriven <-
```

```
  rbind(results_RICLPM_genr_m3_t1w[c(37:38,33:34,81:88),],
        results_RICLPM_genr_m2_dti[c(3:4,7:8,15:18,21:22,29:30),])

results_RICLPM_genr_m3_exploratory <-
  rbind(results_RICLPM_genr_m3_t1w[c(3:32,35:36,39:80),],
        results_RICLPM_genr_m2_dti[c(5:6,9:14,19:20,23:28),])

#Create empty FDR dataframe
fdr_hypothesisdriven_genr_riclpm <-
     matrix(nrow = nrow(results_RICLPM_genr_m3_hypothesisdriven)*4.5, ncol = 3)
fdr_exploratory_genr_riclpm <-
     matrix(nrow = nrow(results_RICLPM_genr_m3_exploratory)*4.5, ncol = 3)

#Bind all pvalues together
pvals_all_hypothesisdriven_genr_riclpm <- c(results_RICLPM_genr_m3_hypothesisdriven[,5],
                                results_RICLPM_genr_m3_hypothesisdriven[,8],
                                results_RICLPM_genr_m3_hypothesisdriven[,11],
                                results_RICLPM_genr_m3_hypothesisdriven[,14],
                                results_RICLPM_genr_m3_hypothesisdriven[,17])
pvals_all_exploratory_genr_riclpm <- c(results_RICLPM_genr_m3_exploratory[,5],
                            results_RICLPM_genr_m3_exploratory[,8],
                            results_RICLPM_genr_m3_exploratory[,11],
                            results_RICLPM_genr_m3_exploratory[,14],
                            results_RICLPM_genr_m3_exploratory[,17])

#Calculate threshold for each test
for(x in 1:nrow(fdr_hypothesisdriven_genr_riclpm)){
  thresh_temp <- x/nrow(fdr_hypothesisdriven_genr_riclpm)*0.05
  if(x == 1){
    thresholds_hypothesisdriven_genr_riclpm <- thresh_temp
  } else {
    thresholds_hypothesisdriven_genr_riclpm <-
      c(thresholds_hypothesisdriven_genr_riclpm, thresh_temp)
  }
}

for(x in 1:nrow(fdr_exploratory_genr_riclpm)){
  thresh_temp <- x/nrow(fdr_exploratory_genr_riclpm)*0.05
  if(x == 1){
    thresholds_exploratory_genr_riclpm <- thresh_temp
  } else {
    thresholds_exploratory_genr_riclpm <-
      c(thresholds_exploratory_genr_riclpm, thresh_temp)
  }
}

#Fill FDR dataframes
fdr_hypothesisdriven_genr_riclpm[,1] <- sort(as.numeric(pvals_all_hypothesisdriven_genr_riclpm))
fdr_hypothesisdriven_genr_riclpm[,2] <- thresholds_hypothesisdriven_genr_riclpm
fdr_hypothesisdriven_genr_riclpm[,3] <-
  ifelse(as.numeric(fdr_hypothesisdriven_genr_riclpm[,1]) <
           as.numeric(fdr_hypothesisdriven_genr_riclpm[,2]), "sig", "nonsig")
```

```
fdr_exploratory_genr_riclpm[,1] <- sort(as.numeric(pvals_all_exploratory_genr_riclpm))
fdr_exploratory_genr_riclpm[,2] <- thresholds_exploratory_genr_riclpm
fdr_exploratory_genr_riclpm[,3] <-
  ifelse(as.numeric(fdr_exploratory_genr_riclpm[,1]) <
           as.numeric(fdr_exploratory_genr_riclpm[,2]), "sig", "nonsig")

#Add correction to tables
#Store significance after FDR-BH as * in
results_RICLPM_genr_m3_hypothesisdriven[,"sig_CS"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_hypothesisdriven[,5]) <= 0.00601851851851852, "*", "")
results_RICLPM_genr_m3_hypothesisdriven[,"sig_AR_CBCL"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_hypothesisdriven[,8]) <= 0.00601851851851852, "*", "")
results_RICLPM_genr_m3_hypothesisdriven[,"sig_AR_MRI"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_hypothesisdriven[,11]) <= 0.00601851851851852, "*", "")
results_RICLPM_genr_m3_hypothesisdriven[,"sig_CL_CBCLMRI"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_hypothesisdriven[,14]) <= 0.00601851851851852, "*", "")
results_RICLPM_genr_m3_hypothesisdriven[,"sig_CL_MRICBCL"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_hypothesisdriven[,17]) <= 0.00601851851851852, "*", "")

results_RICLPM_genr_m3_exploratory[,"sig_CS"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_exploratory[,5]) <= 0.0150617283950617, "*", "")
results_RICLPM_genr_m3_exploratory[,"sig_AR_CBCL"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_exploratory[,8]) <= 0.0150617283950617, "*", "")
results_RICLPM_genr_m3_exploratory[,"sig_AR_MRI"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_exploratory[,11]) <= 0.0150617283950617, "*", "")
results_RICLPM_genr_m3_exploratory[,"sig_CL_CBCLMRI"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_exploratory[,14]) <= 0.0150617283950617, "*", "")
results_RICLPM_genr_m3_exploratory[,"sig_CL_MRICBCL"] <-
  ifelse(as.numeric(results_RICLPM_genr_m3_exploratory[,17]) <= 0.0150617283950617, "*", "")


#Print tables
#Hypothesis driven
results_RICLPM_genr_m3_hypothesisdriven[,c(3:17)] <-
  sapply(results_RICLPM_genr_m3_hypothesisdriven[,c(3:17)], as.numeric)
results_RICLPM_genr_m3_hypothesisdriven[,c(1:5,18,6:8,19,9:11,20,12:14,21,15:17,22)] %>%
  kable(digits = 2,
        format = "latex",
      caption="Results hypothesis driven RICLPM Generation R",
      col.names = c("Brain region", "Timepoint",
                    "CS B", "SE", "p", "",
                    "AR CBCL B", "SE", "p", "",
                    "AR MRI B", "SE", "p", "",
                    "CL CBCL>MRI B", "SE", "p", "",
                    "CL MRI>CBCL B", "SE", "p", ""),
      align="r") %>%
  kable_classic(full_width = F, html_font = "helvetica")
```

```
#Exploratory
results_RICLPM_genr_m3_exploratory[,c(3:17)] <-
  sapply(results_RICLPM_genr_m3_exploratory[,c(3:17)], as.numeric)
results_RICLPM_genr_m3_exploratory[,c(1:5,18,6:8,19,9:11,20,12:14,21,15:17,22)] %>%
  kable(digits = 2,
        format = "latex",
```

Table 3: Results hypothesis driven RICLPM Generation R

	Brain region	Timepoint	CS B	SE	p		AR CBCL B	SE	p		AR MRI B	SE	p		CL CBCL>MRI B	SE	p		CL MRI>CBCL B	SE	p
37	medialorbitofrontal_vol	T1-T2	0.09	0.05	0.10		0.44	0.18	0.01		0.36	0.19	0.06		0.05	0.07	0.49		0.08	0.10	0.43
38	NA	T2-T3	NA	NA	NA	NA	0.48	0.14	0.00	*	0.29	0.13	0.02		0.05	0.07	0.42		0.01	0.07	0.83
33	lateralorbitofrontal_vol	T1-T2	0.03	0.05	0.50		0.52	0.16	0.00	*	0.35	0.21	0.09		-0.01	0.06	0.86		0.06	0.10	0.57
34	NA	T2-T3	NA	NA	NA	NA	0.54	0.13	0.00	*	0.32	0.10	0.00	*	0.06	0.06	0.25		-0.09	0.07	0.24
81	Left_Amygdala_vol	T1-T2	-0.03	0.07	0.65		0.43	0.16	0.01		-0.05	0.19	0.81		0.03	0.11	0.77		-0.02	0.11	0.88
82	NA	T2-T3	NA	NA	NA	NA	0.47	0.15	0.00	*	0.42	0.16	0.01		0.05	0.08	0.52		0.04	0.09	0.67
83	Right_Amygdala_vol	T1-T2	0.00	0.06	0.99		0.44	0.16	0.01		-0.30	0.28	0.28		0.05	0.12	0.68		0.05	0.12	0.70
84	NA	T2-T3	NA	NA	NA	NA	0.47	0.15	0.00	*	0.38	0.11	0.00	*	0.09	0.07	0.18		0.03	0.08	0.69
85	Left_Accumbens_area_vol	T1-T2	-0.04	0.08	0.64		0.48	0.15	0.00	*	0.09	0.19	0.63		0.11	0.10	0.27		0.10	0.17	0.58
86	NA	T2-T3	NA	NA	NA	NA	0.50	0.14	0.00	*	0.07	0.16	0.68		0.17	0.11	0.12		0.02	0.08	0.77
87	Right_Accumbens_area_vol	T1-T2	0.00	0.06	0.94		0.45	0.16	0.01	*	0.25	0.18	0.16		0.13	0.09	0.16		0.12	0.14	0.39
88	NA	T2-T3	NA	NA	NA	NA	0.47	0.15	0.00	*	0.21	0.16	0.18		0.09	0.10	0.33		0.04	0.08	0.62
3	cgc_FA	T1-T2	0.02	0.06	0.76		0.30	0.14	0.04		-0.31	0.11	0.00	*	-0.07	0.11	0.54		-0.03	0.11	0.79
4	NA	T2-T3	NA	NA	NA	NA	0.36	0.15	0.02		-0.19	0.15	0.20		-0.03	0.12	0.83		-0.06	0.10	0.57
7	unc_FA	T1-T2	0.03	0.06	0.66		0.28	0.14	0.05		-0.36	0.11	0.00	*	0.17	0.09	0.08		0.02	0.12	0.84
8	NA	T2-T3	NA	NA	NA	NA	0.35	0.14	0.01		-0.34	0.24	0.15		0.01	0.15	0.96		0.05	0.14	0.74
15	fmi_FA	T1-T2	-0.03	0.06	0.61		0.26	0.14	0.06		-0.30	0.14	0.03		0.05	0.10	0.63		0.06	0.13	0.64
16	NA	T2-T3	NA	NA	NA	NA	0.30	0.18	0.08		-0.13	0.26	0.62		0.26	0.19	0.17		0.06	0.16	0.70
17	cgc_MD	T1-T2	0.01	0.06	0.91		0.32	0.13	0.02		-0.22	0.16	0.16		0.15	0.13	0.22		0.10	0.14	0.50
18	NA	T2-T3	NA	NA	NA	NA	0.35	0.15	0.02		0.16	0.10	0.12		0.25	0.09	0.01		0.08	0.10	0.40
21	unc_MD	T1-T2	-0.03	0.06	0.67		0.27	0.15	0.06		-0.32	0.39	0.42		-0.07	0.17	0.69		0.07	0.24	0.76
22	NA	T2-T3	NA	NA	NA	NA	0.33	0.16	0.03		0.46	0.09	0.00	*	0.01	0.08	0.89		-0.08	0.09	0.39
29	fmi_MD	T1-T2	0.08	0.07	0.21		0.28	0.14	0.05		-0.11	0.13	0.40		0.01	0.13	0.96		0.12	0.09	0.20
30	NA	T2-T3	NA	NA	NA	NA	0.36	0.16	0.02		0.08	0.13	0.56		0.09	0.16	0.58		0.03	0.10	0.78

```
      caption="Results exploratory RICLPM Generation R",
      col.names = c("Brain region", "Timepoint",
                    "CS B", "SE", "p", "",
                    "AR CBCL B", "SE", "p", "",
                    "AR MRI B", "SE", "p", "",
                    "CL CBCL>MRI B", "SE", "p", "",
                    "CL MRI>CBCL B", "SE", "p", ""),
      align="r") %>%
  kable_classic(full_width = F, html_font = "helvetica")
```

```
#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_RICLPM_genr_m3_hypothesisdriven, "results_RICLPM_genr_m3_hypothesisdriven.csv")
write.csv(results_RICLPM_genr_m3_exploratory, "results_RICLPM_genr_m3_exploratory.csv")
```

2.1.10 Model fit comparison CLPM/RI-CLPM

```
#Model comparison
#Redo each model and compare model fit between CLPM and RI-CLPM in Generation R
#Create empty dataframe to store results comparison model fits
results_modelfit_comparison <- data.frame()

#Reset rowcount to keep track of where we are in the loop
rowcount <- 1

#Specify the model for all brain morphology measures
#T1w
for(x in structs3_genr[1:43]){
  #Rerun CLPMs
  CLPM_genr <- paste0(
    '
    # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",
```

Table 4: Results exploratory RICLPM Generation R

	Brain region	Timepoint	CS B	SE	p		AR CBCL B	SE	p		AR MRI B	SE	p		CL CBCL>MRI B	SE	p		CL MRI>CBCL B	SE	p
3	Thalamus_Proper_vol	T1-T2	0.02	0.04	0.56		0.43	0.17	0.01	*	0.32	0.11	0.00	*	0.00	0.05	0.97		0.06	0.09	0.54
4	NA	T2-T3	NA	NA	NA	NA	0.47	0.15	0.00	*	0.42	0.12	0.00	*	-0.05	0.05	0.25		0.02	0.08	0.78
5	Caudate_vol	T1-T2	0.00	0.05	1.00		0.50	0.16	0.00	*	0.95	0.09	0.00	*	-0.03	0.03	0.33		0.01	0.12	0.95
6	NA	T2-T3	NA	NA	NA	NA	0.51	0.14	0.00	*	0.77	0.07	0.00	*	-0.02	0.02	0.46		-0.10	0.09	0.30
7	Putamen_vol	T1-T2	0.07	0.08	0.38		0.44	0.16	0.01	*	0.71	0.20	0.00	*	-0.02	0.06	0.79		0.13	0.16	0.42
8	NA	T2-T3	NA	NA	NA	NA	0.49	0.13	0.00	*	0.70	0.12	0.00	*	0.07	0.04	0.07		-0.02	0.12	0.86
9	Pallidum_vol	T1-T2	0.09	0.06	0.17		0.53	0.16	0.00	*	0.10	0.21	0.64		0.05	0.10	0.60		0.12	0.12	0.33
10	NA	T2-T3	NA	NA	NA	NA	0.55	0.14	0.00	*	0.33	0.11	0.00	*	-0.08	0.06	0.19		-0.03	0.06	0.63
11	Hippocampus_vol	T1-T2	0.00	0.05	0.92		0.48	0.16	0.00	*	0.10	0.16	0.54		-0.05	0.05	0.38		0.06	0.11	0.56
12	NA	T2-T3	NA	NA	NA	NA	0.51	0.14	0.00	*	0.45	0.12	0.00	*	-0.02	0.03	0.65		-0.01	0.09	0.90
13	bankssts_vol	T1-T2	-0.07	0.06	0.25		0.46	0.16	0.00	*	0.33	0.26	0.20		0.00	0.07	0.98		0.05	0.11	0.62
14	NA	T2-T3	NA	NA	NA	NA	0.49	0.15	0.00	*	0.68	0.10	0.00	*	0.03	0.03	0.41		-0.02	0.10	0.81
15	caudalanteriorcingulate_vol	T1-T2	-0.01	0.05	0.83		0.43	0.16	0.01	*	-0.04	0.55	0.93		0.05	0.08	0.51		0.07	0.14	0.64
16	NA	T2-T3	NA	NA	NA	NA	0.47	0.15	0.00	*	0.68	0.10	0.00	*	-0.02	0.04	0.62		0.02	0.13	0.87
17	caudalmiddlefrontal_vol	T1-T2	-0.01	0.06	0.82		0.45	0.16	0.00	*	0.34	0.24	0.16		0.09	0.08	0.26		0.05	0.12	0.67
18	NA	T2-T3	NA	NA	NA	NA	0.48	0.14	0.00	*	0.64	0.10	0.00	*	0.02	0.04	0.65		-0.03	0.10	0.73
19	cuneus_vol	T1-T2	-0.03	0.04	0.48		0.43	0.16	0.01	*	-0.67	0.92	0.46		-0.08	0.12	0.50		0.08	0.21	0.71
20	NA	T2-T3	NA	NA	NA	NA	0.46	0.15	0.00	*	0.48	0.15	0.00	*	0.03	0.04	0.35		0.11	0.18	0.54
21	entorhinal_vol	T1-T2	-0.03	0.06	0.54		0.35	0.17	0.04		0.17	0.15	0.25		-0.06	0.10	0.53		0.06	0.14	0.68
22	NA	T2-T3	NA	NA	NA	NA	0.41	0.15	0.01	*	-0.04	0.21	0.87		0.00	0.11	0.99		0.24	0.15	0.12
23	fusiform_vol	T1-T2	-0.14	0.07	0.04		0.41	0.16	0.01	*	0.33	0.20	0.10		-0.10	0.10	0.35		0.06	0.10	0.55
24	NA	T2-T3	NA	NA	NA	NA	0.44	0.15	0.00	*	0.67	0.07	0.00	*	-0.01	0.05	0.87		0.00	0.09	0.98
25	inferiorparietal_vol	T1-T2	-0.03	0.08	0.71		0.43	0.16	0.01	*	0.78	0.13	0.00	*	0.01	0.05	0.88		0.03	0.09	0.71
26	NA	T2-T3	NA	NA	NA	NA	0.47	0.14	0.00	*	0.88	0.05	0.00	*	0.05	0.04	0.15		-0.02	0.07	0.82
27	inferiortemporal_vol	T1-T2	-0.05	0.07	0.47		0.47	0.16	0.00	*	0.35	0.16	0.03		-0.04	0.07	0.61		-0.03	0.10	0.76
28	NA	T2-T3	NA	NA	NA	NA	0.48	0.14	0.00	*	0.49	0.09	0.00	*	-0.07	0.05	0.16		-0.10	0.10	0.32
29	isthmuscingulate_vol	T1-T2	0.00	0.05	0.92		0.44	0.16	0.01	*	0.58	0.25	0.02		0.03	0.05	0.47		0.05	0.12	0.67
30	NA	T2-T3	NA	NA	NA	NA	0.47	0.15	0.00	*	0.65	0.10	0.00	*	0.03	0.03	0.31		-0.02	0.10	0.83
31	lateraloccipital_vol	T1-T2	-0.01	0.04	0.85		0.43	0.16	0.01	*	0.01	0.34	0.98		0.02	0.07	0.81		0.07	0.13	0.59
32	NA	T2-T3	NA	NA	NA	NA	0.47	0.15	0.00	*	0.52	0.12	0.00	*	0.06	0.04	0.08		0.02	0.11	0.84
35	lingual_vol	T1-T2	-0.09	0.04	0.03		0.43	0.16	0.01	*	-0.86	0.82	0.30		-0.22	0.19	0.23		0.10	0.17	0.57
36	NA	T2-T3	NA	NA	NA	NA	0.46	0.16	0.00	*	0.59	0.17	0.00	*	0.04	0.04	0.31		0.14	0.16	0.39
39	middletemporal_vol	T1-T2	0.00	0.06	0.99		0.45	0.16	0.00	*	0.14	0.15	0.37		0.01	0.08	0.88		0.06	0.09	0.51
40	NA	T2-T3	NA	NA	NA	NA	0.48	0.14	0.00	*	0.45	0.13	0.00	*	0.05	0.06	0.36		-0.05	0.08	0.56
41	parahippocampal_vol	T1-T2	-0.03	0.04	0.42		0.44	0.16	0.01	*	0.46	0.26	0.07		-0.05	0.07	0.52		0.12	0.17	0.49
42	NA	T2-T3	NA	NA	NA	NA	0.47	0.14	0.00	*	0.41	0.09	0.00	*	-0.01	0.05	0.77		-0.02	0.12	0.88
43	paracentral_vol	T1-T2	-0.03	0.06	0.66		0.44	0.16	0.01	*	-0.29	0.72	0.69		-0.01	0.12	0.96		0.04	0.13	0.73
44	NA	T2-T3	NA	NA	NA	NA	0.48	0.15	0.00	*	0.55	0.13	0.00	*	0.01	0.04	0.84		0.07	0.10	0.50
45	parsopercularis_vol	T1-T2	-0.07	0.06	0.24		0.44	0.16	0.01	*	0.14	0.30	0.64		-0.07	0.09	0.46		0.05	0.11	0.69
46	NA	T2-T3	NA	NA	NA	NA	0.48	0.15	0.00	*	0.57	0.10	0.00	*	0.00	0.05	0.99		-0.02	0.09	0.82
47	parsorbitalis_vol	T1-T2	-0.03	0.08	0.68		0.52	0.14	0.00	*	0.34	0.17	0.05		-0.10	0.08	0.20		0.01	0.10	0.94
48	NA	T2-T3	NA	NA	NA	NA	0.54	0.13	0.00	*	0.63	0.10	0.00	*	0.09	0.05	0.08		-0.10	0.08	0.21
49	parstriangularis_vol	T1-T2	-0.04	0.08	0.65		0.54	0.14	0.00	*	0.08	0.26	0.77		-0.09	0.10	0.36		0.07	0.15	0.64
50	NA	T2-T3	NA	NA	NA	NA	0.55	0.13	0.00	*	0.75	0.10	0.00	*	0.05	0.04	0.22		-0.08	0.13	0.54
51	pericalcarine_vol	T1-T2	-0.08	0.07	0.25		0.37	0.16	0.02		0.77	0.13	0.00	*	-0.05	0.07	0.50		0.08	0.13	0.56
52	NA	T2-T3	NA	NA	NA	NA	0.41	0.15	0.01	*	0.58	0.09	0.00	*	0.04	0.06	0.46		0.13	0.11	0.26
53	postcentral_vol	T1-T2	0.00	0.05	0.94		0.43	0.17	0.01	*	-0.24	0.69	0.73		0.03	0.08	0.74		0.03	0.14	0.81
54	NA	T2-T3	NA	NA	NA	NA	0.47	0.15	0.00	*	0.62	0.11	0.00	*	0.00	0.03	0.92		0.00	0.12	0.97
55	posteriorcingulate_vol	T1-T2	-0.13	0.07	0.07		0.47	0.15	0.00	*	0.56	0.36	0.13		-0.06	0.09	0.50		0.04	0.10	0.70
56	NA	T2-T3	NA	NA	NA	NA	0.48	0.15	0.00	*	0.82	0.05	0.00	*	0.01	0.03	0.67		0.00	0.08	0.99
57	precentral_vol	T1-T2	0.02	0.05	0.76		0.41	0.17	0.02		-0.04	0.56	0.94		0.08	0.09	0.36		0.03	0.11	0.76
58	NA	T2-T3	NA	NA	NA	NA	0.45	0.15	0.00	*	0.63	0.10	0.00	*	0.02	0.04	0.72		-0.02	0.09	0.80
59	precuneus_vol	T1-T2	0.05	0.06	0.42		0.46	0.16	0.00	*	0.77	0.12	0.00	*	0.03	0.04	0.55		0.05	0.09	0.57
60	NA	T2-T3	NA	NA	NA	NA	0.49	0.14	0.00	*	0.80	0.07	0.00	*	0.05	0.03	0.08		-0.03	0.07	0.69
61	rostralanteriorcingulate_vol	T1-T2	0.07	0.07	0.31		0.47	0.16	0.00	*	0.33	0.30	0.27		0.12	0.08	0.15		0.05	0.10	0.58
62	NA	T2-T3	NA	NA	NA	NA	0.49	0.14	0.00	*	0.71	0.07	0.00	*	0.00	0.04	0.91		0.01	0.08	0.91
63	rostralmiddlefrontal_vol	T1-T2	0.02	0.07	0.74		0.48	0.16	0.00	*	0.13	0.24	0.58		0.04	0.08	0.65		0.02	0.09	0.80
64	NA	T2-T3	NA	NA	NA	NA	0.51	0.15	0.00	*	0.72	0.08	0.00	*	0.05	0.05	0.28		-0.03	0.08	0.69
65	superiorfrontal_vol	T1-T2	-0.05	0.08	0.57		0.46	0.15	0.00	*	0.64	0.15	0.00	*	0.07	0.06	0.24		0.01	0.08	0.87
66	NA	T2-T3	NA	NA	NA	NA	0.49	0.14	0.00	*	0.83	0.06	0.00	*	0.00	0.04	0.96		0.01	0.07	0.87
67	superiorparietal_vol	T1-T2	0.08	0.08	0.30		0.39	0.16	0.01	*	0.74	0.12	0.00	*	0.07	0.06	0.24		0.03	0.10	0.74
68	NA	T2-T3	NA	NA	NA	NA	0.45	0.14	0.00	*	0.77	0.11	0.00	*	0.14	0.06	0.01	*	0.00	0.08	0.98
69	superiortemporal_vol	T1-T2	-0.02	0.08	0.80		0.49	0.16	0.00	*	0.56	0.14	0.00	*	-0.05	0.06	0.43		0.02	0.09	0.82
70	NA	T2-T3	NA	NA	NA	NA	0.51	0.14	0.00	*	0.63	0.08	0.00	*	0.02	0.05	0.73		-0.10	0.08	0.23
71	supramarginal_vol	T1-T2	-0.02	0.06	0.76		0.44	0.16	0.01	*	0.87	0.12	0.00	*	-0.04	0.05	0.43		0.03	0.09	0.75
72	NA	T2-T3	NA	NA	NA	NA	0.48	0.14	0.00	*	0.82	0.07	0.00	*	0.04	0.03	0.14		-0.03	0.08	0.66
73	frontalpole_vol	T1-T2	0.05	0.08	0.51		0.40	0.18	0.02		0.11	0.14	0.42		0.01	0.13	0.94		0.09	0.12	0.47
74	NA	T2-T3	NA	NA	NA	NA	0.46	0.15	0.00	*	0.19	0.13	0.15		0.07	0.13	0.60		0.02	0.10	0.85
75	temporalpole_vol	T1-T2	0.07	0.08	0.39		0.40	0.18	0.03		-0.08	0.15	0.60		-0.10	0.15	0.51		0.04	0.12	0.73
76	NA	T2-T3	NA	NA	NA	NA	0.47	0.15	0.00	*	0.23	0.11	0.03		0.04	0.11	0.72		0.07	0.08	0.39
77	transversetemporal_vol	T1-T2	-0.08	0.06	0.19		0.49	0.15	0.00	*	0.97	0.17	0.00	*	0.04	0.04	0.33		0.04	0.15	0.81
78	NA	T2-T3	NA	NA	NA	NA	0.51	0.14	0.00	*	0.70	0.07	0.00	*	0.02	0.04	0.57		-0.06	0.10	0.55
79	insula_vol	T1-T2	-0.10	0.06	0.09		0.47	0.16	0.00	*	0.58	0.13	0.00	*	-0.07	0.06	0.27		0.05	0.09	0.62
80	NA	T2-T3	NA	NA	NA	NA	0.50	0.15	0.00	*	0.37	0.10	0.00	*	-0.02	0.06	0.73		-0.04	0.07	0.60
510	cst_FA	T1-T2	0.00	0.07	0.96		0.33	0.13	0.01	*	-0.29	0.23	0.20		0.14	0.14	0.31		-0.02	0.17	0.90
610	NA	T2-T3	NA	NA	NA	NA	0.38	0.15	0.01	*	0.32	0.16	0.05		-0.15	0.12	0.21		0.00	0.13	1.00
91	ilf_FA	T1-T2	0.05	0.06	0.35		0.27	0.15	0.06		-0.40	0.13	0.00	*	0.09	0.08	0.24		0.06	0.16	0.73
101	NA	T2-T3	NA	NA	NA	NA	0.36	0.15	0.02		-0.52	0.22	0.02		0.02	0.20	0.91		-0.06	0.15	0.71
111	slf_FA	T1-T2	0.07	0.06	0.27		0.29	0.15	0.05		-0.50	0.17	0.00	*	0.08	0.11	0.43		0.00	0.15	1.00
121	NA	T2-T3	NA	NA	NA	NA	0.35	0.15	0.02		-0.19	0.19	0.34		0.01	0.17	0.95		-0.04	0.13	0.79
131	fma_FA	T1-T2	0.06	0.05	0.23		0.27	0.15	0.07		-0.32	0.14	0.02		0.09	0.10	0.37		-0.01	0.12	0.94
141	NA	T2-T3	NA	NA	NA	NA	0.33	0.15	0.03		-0.34	0.18	0.05		-0.06	0.17	0.72		0.04	0.11	0.73
191	cst_MD	T1-T2	-0.04	0.09	0.68		0.32	0.13	0.01	*	0.22	0.18	0.21		0.02	0.16	0.90		-0.03	0.08	0.73
201	NA	T2-T3	NA	NA	NA	NA	0.37	0.14	0.01	*	0.49	0.14	0.00	*	0.23	0.13	0.07		-0.07	0.09	0.46
231	ilf_MD	T1-T2	0.03	0.05	0.59		0.28	0.14	0.05		0.08	0.15	0.62		0.09	0.09	0.31		0.09	0.15	0.54
241	NA	T2-T3	NA	NA	NA	NA	0.34	0.15	0.03		0.14	0.20	0.49		0.04	0.10	0.71		0.05	0.21	0.82
251	slf_MD	T1-T2	-0.01	0.04	0.87		0.28	0.14	0.05		0.03	0.13	0.83		0.01	0.08	0.91		0.08	0.18	0.67
261	NA	T2-T3	NA	NA	NA	NA	0.34	0.15	0.02		0.00	0.24	0.99		0.08	0.09	0.40		-0.12	0.23	0.61
271	fma_MD	T1-T2	0.00	0.05	0.94		0.27	0.14	0.06		-0.04	0.18	0.83		-0.03	0.11	0.77		0.08	0.11	0.48
281	NA	T2-T3	NA	NA	NA	NA	0.34	0.15	0.03		0.05	0.16	0.77		0.20	0.13	0.13		-0.15	0.10	0.14

```
  #Estimate time independent predictors
  ' \n \n sum_dp_5_sqrt_scaled ~ sex +  maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch +
  child_nationalorigin_wes \n ',
  x,"_f05_scaled", ' ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch +
  child_nationalorigin_wes \n

  #Estimate time dependent predictors
  sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
  sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
  sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
  x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled \n ',
  x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled \n ',
  x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
  x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)


#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
CLPM_genr_fit <-
  lavaan(CLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                               survey.design = survey_design_genr)

#Rerun RICLPMs
RICLPM_genr <-
  paste0('
         # Create between components (random intercepts)
         RIx =~ 1*sum_dp_5_sqrt_scaled + 1*sum_dp_9m_sqrt_scaled
         + 1*sum_dp_14_sqrt_scaled
         RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled",
         ' \n + 1*', x,"_f13_scaled",

         # Create within-person centered variables
```

```
          ' \n  wx1 =~ 1*sum_dp_5_sqrt_scaled
          wx2 =~ 1*sum_dp_9m_sqrt_scaled
          wx3 =~ 1*sum_dp_14_sqrt_scaled
          wy1 =~ 1* \n', x,"_f05_scaled",
          ' \n wy2 =~ 1* \n', x,"_f09_scaled",
          ' \n wy3 =~ 1* \n', x,"_f13_scaled",

          # Estimate the lagged effects between the variables
          ' \n wx2 + wy2 ~ wx1 + wy1
          wx3 + wy3 ~ wx2 + wy2

          #Estimate time independent predictors
          RIx ~ sex + maternal_education_middle +
          maternal_education_high + child_nationalorigin_dutch +
          child_nationalorigin_wes
          RIy ~ sex + maternal_education_middle +
          maternal_education_high + child_nationalorigin_dutch +
          child_nationalorigin_wes

          #Estimate time dependent predictors
          wx1 ~ agechild_GR1075
          wx2 ~ AgeChild_CBCL9m
          wx3 ~ AGECHILD_GR1093
          wy1 ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled
          wy2 ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled
          wy3 ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled

          # Estimate the covariance between variables at the first wave.
          wx1 ~~ wy1 # Covariance

          # Estimate the covariances between the residuals of variables
          wx2 ~~ wy2
          wx3 ~~ wy3

          # Estimate the variance and covariance of the random intercepts.
          RIx ~~ RIx
          RIy ~~ RIy
          RIx ~~ RIy

          # Estimate the (residual) variance of the within-person centered variables.
          wx1 ~~ wx1 # Variances
          wy1 ~~ wy1
          wx2 ~~ wx2 # Residual variances
          wy2 ~~ wy2
          wx3 ~~ wx3
          wy3 ~~ wy3
          ')

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
RICLPM_genr_fit <-
```

```
    lavaan(RICLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                   survey.design = survey_design_genr)

  #Perform model fit comparison using ANOVA
  comparison <- anova(RICLPM_genr_CSA, CLPM_genr_CSA)

  #Store results ANOVA to dataframe
  results_modelfit_comparison[c(rowcount:(rowcount+1)),1] <- x
  results_modelfit_comparison[c(rowcount:(rowcount+1)),c(2:8)] <-
    comparison[,1:7]

  #Update rowcount
  rowcount <- rowcount + 2
}

#DTI
for(x in structs3_genr[44:57]){
  #Rerun CLPMs
  CLPM_genr <- paste0(
    '
    # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_5_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch +
    child_nationalorigin_wes \n ',
    x,"_f05_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch +
    child_nationalorigin_wes \n

    #Estimate time dependent predictors
    sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
    sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
    x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
```

```
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)


#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

#Fit the model specified above
CLPM_genr_fit <-
  lavaan(CLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                               survey.design = survey_design_genr)

#Rerun RICLPMs
RICLPM_genr <-
  paste0('
         # Create between components (random intercepts)
         RIx =~ 1*sum_dp_5_sqrt_scaled + 1*sum_dp_9m_sqrt_scaled
         + 1*sum_dp_14_sqrt_scaled
         RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled",
         ' \n + 1*', x,"_f13_scaled",

         # Create within-person centered variables
         ' \n  wx1 =~ 1*sum_dp_5_sqrt_scaled
         wx2 =~ 1*sum_dp_9m_sqrt_scaled
         wx3 =~ 1*sum_dp_14_sqrt_scaled
         wy1 =~ 1* \n', x,"_f05_scaled",
         ' \n wy2 =~ 1* \n', x,"_f09_scaled",
         ' \n wy3 =~ 1* \n', x,"_f13_scaled",

         # Estimate the lagged effects between the variables
         ' \n wx2 + wy2 ~ wx1 + wy1
         wx3 + wy3 ~ wx2 + wy2

         #Estimate time independent predictors
         RIx ~ sex + maternal_education_middle +
         maternal_education_high + child_nationalorigin_dutch +
         child_nationalorigin_wes
         RIy ~ sex + maternal_education_middle +
         maternal_education_high + child_nationalorigin_dutch +
         child_nationalorigin_wes

         #Estimate time dependent predictors
         wx1 ~ agechild_GR1075
         wx2 ~ AgeChild_CBCL9m
         wx3 ~ AGECHILD_GR1093
         wy1 ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled
         wy2 ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled
         wy3 ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled
```

```
          # Estimate the covariance between variables at the first wave.
          wx1 ~~ wy1 # Covariance

          # Estimate the covariances between the residuals of variables
          wx2 ~~ wy2
          wx3 ~~ wy3

          # Estimate the variance and covariance of the random intercepts.
          RIx ~~ RIx
          RIy ~~ RIy
          RIx ~~ RIy

          # Estimate the (residual) variance of the within-person centered variables.
          wx1 ~~ wx1 # Variances
          wy1 ~~ wy1
          wx2 ~~ wx2 # Residual variances
          wy2 ~~ wy2
          wx3 ~~ wx3
          wy3 ~~ wy3
          ')

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

  #Fit the model specified above
  RICLPM_genr_fit <-
    lavaan(RICLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                   survey.design = survey_design_genr)

  #Perform model fit comparison using ANOVA
  comparison <- anova(RICLPM_genr_CSA, CLPM_genr_CSA)

  #Store results ANOVA to dataframe
  results_modelfit_comparison[c(rowcount:(rowcount+1)),1] <- x
  results_modelfit_comparison[c(rowcount:(rowcount+1)),c(2:8)] <-
    comparison[,1:7]

  #Update rowcount
  rowcount <- rowcount + 2
}

#Save model comparisons
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_modelfit_comparison, "results_modelfit_comparison.csv",
          row.names = F, quote = F)
```

2.1.11 CLPM with 2 time-points (meta-analysis)

This part is run to accommodate the final sensitivity analysis in which we meta-analyze results from Generation R and the ABCD-study.

```
#Rerun CLPM using two time-points only in Generation R
##Make list of imputed datasets to feed to lavaan
implist_genr <- lapply(seq(dsImp_genr$m), function(im) complete(dsImp_genr, im))

##Create dummy variables for categorical variables
for(x in seq(dsImp_genr$m)){
  #Sex
  implist_genr[[x]]$sex <-
    ifelse(implist_genr[[x]]$GENDER == "girl", 1, 0)
  #Maternal education
  implist_genr[[x]]$maternal_education_high <-
    ifelse(implist_genr[[x]]$maternal_education == "high", 1, 0)
  implist_genr[[x]]$maternal_education_middle <-
    ifelse(implist_genr[[x]]$maternal_education == "middle", 1, 0)
  #Child national origin
  implist_genr[[x]]$child_nationalorigin_dutch <-
    ifelse(implist_genr[[x]]$child_nationalorigin == "Dutch", 1, 0)
  implist_genr[[x]]$child_nationalorigin_wes <-
    ifelse(implist_genr[[x]]$child_nationalorigin == "other western", 1, 0)
}

#Subset data based on complete T2 and T3 data
#Split T1-weighted and DTI sample
#Do this so that there are no missings in each dataframe
implist_t1w_genr <- list()
implist_dti_genr <- list()


for(x in seq(dsImp_genr$m)){
  implist_t1w_genr[[x]] <-
    subset(implist_genr[[x]], avail_f9 == 1 & avail_f13 == 1 &
           !is.na(sum_dp_9m_sqrt_scaled) & !is.na(sum_dp_14_sqrt_scaled))
}

for(x in seq(dsImp_genr$m)){
  implist_dti_genr[[x]] <-
    subset(implist_genr[[x]], avail_f9_dti == 1 & avail_f13_dti == 1 &
           !is.na(sum_dp_9m_sqrt_scaled) & !is.na(sum_dp_14_sqrt_scaled))
}

#Create imputationlist object
data_genr_t1w <- imputationList(implist_t1w_genr)
data_genr_dti <- imputationList(implist_dti_genr)

#Create empty dataframes to store results
results_CLPM_2tp_genr_m1_t1w <- data.frame()
fitmeaures_CLPM_2tp_genr_m1_t1w <- data.frame()
rsquared_CLPM_2tp_genr_m1_t1w <- data.frame()
results_CLPM_2tp_genr_m2_t1w <- data.frame()
fitmeaures_CLPM_2tp_genr_m2_t1w <- data.frame()
rsquared_CLPM_2tp_genr_m2_t1w <- data.frame()
results_CLPM_2tp_genr_m3_t1w <- data.frame()
fitmeaures_CLPM_2tp_genr_m3_t1w <- data.frame()
```

```
rsquared_CLPM_2tp_genr_m3_t1w <- data.frame()

results_CLPM_2tp_genr_m1_dti <- data.frame()
fitmeaures_CLPM_2tp_genr_m1_dti <- data.frame()
rsquared_CLPM_2tp_genr_m1_dti <- data.frame()
results_CLPM_2tp_genr_m2_dti <- data.frame()
fitmeaures_CLPM_2tp_genr_m2_dti <- data.frame()
rsquared_CLPM_2tp_genr_m2_dti <- data.frame()


#Specify rowcount to keep track of where we are in the loop
rowcount_clpm_2tp <- 1
rowcount_fit <- 1

#Model 1
#T1w
#Specify the model for all brain morphology measures
for(x in structs3_genr[1:43]){
  CLPM_2tp_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_9m_sqrt_scaled ~ sex \n ',
    x,"_f09_scaled", ' ~ sex + maternal_education_middle \n

    #Estimate time dependent predictors
    sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Variances
    x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ', # Residual variances
    x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

  #Fit the model specified above
  CLPM_2tp_genr_fit <- lavaan(CLPM_2tp_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
```

```
  CLPM_2tp_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_2tp_genr_fit,
                                     survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_2tp_genr_m1_t1w[(rowcount_clpm_2tp+2),1] <- x
  #Cross-sectional associations T1
  results_CLPM_2tp_genr_m1_t1w[c((rowcount_clpm_2tp+2)),2:4] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_2tp_genr_m1_t1w[c((rowcount_clpm_2tp+2)),5:7] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1),c(5,6,8)]
  #MRI
  results_CLPM_2tp_genr_m1_t1w[c((rowcount_clpm_2tp+2)),8:10] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_2tp_genr_m1_t1w[c((rowcount_clpm_2tp+2)),11:13] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_2tp_genr_m1_t1w[c((rowcount_clpm_2tp+2)),14:16] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_2tp_genr_m1_t1w[1,c(3,6,9,12,15)] <-
     c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_2tp_genr_m1_t1w[2,] <- c("Brain region", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value","B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_2tp_genr_m1_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_2tp_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_2tp_genr_m1_t1w[rowcount_fit, 1:5] <- c(x, lavInspect(CLPM_2tp_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm_2tp <- rowcount_clpm_2tp + 1
  rowcount_fit <- rowcount_fit + 1
}


#Reset rowcount to keep track of where we are in the loop
rowcount_clpm_2tp <- 1
rowcount_fit <- 1

#DTI
#Specify the model for all brain morphology measures
for(x in structs3_genr[44:57]){
  CLPM_2tp_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",
```

```
  #Estimate time independent predictors
  ' \n \n sum_dp_9m_sqrt_scaled ~ sex \n ',
  x,"_f09_scaled", ' ~ sex + maternal_education_middle \n

  #Estimate time dependent predictors
  sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
  sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
  x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
  x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ', # Residual variances
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

#Fit the model specified above
CLPM_2tp_genr_fit <- lavaan(CLPM_2tp_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_2tp_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_2tp_genr_fit,
                           survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_CLPM_2tp_genr_m1_dti[(rowcount_clpm_2tp+2),1] <- x
#Cross-sectional associations T1
results_CLPM_2tp_genr_m1_dti[c((rowcount_clpm_2tp+2)),2:4] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_2tp_genr_m1_dti[c((rowcount_clpm_2tp+2)),5:7] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1),c(5,6,8)]
#MRI
results_CLPM_2tp_genr_m1_dti[c((rowcount_clpm_2tp+2)),8:10] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4),c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_CLPM_2tp_genr_m1_dti[c((rowcount_clpm_2tp+2)),11:13] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3),c(5,6,8)]
#MRI -> CBCL
results_CLPM_2tp_genr_m1_dti[c((rowcount_clpm_2tp+2)),14:16] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2),c(5,6,8)]
#Which measure (actually only needs to be specified once)
results_CLPM_2tp_genr_m1_dti[1,c(3,6,9,12,15)] <-
```

```
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_2tp_genr_m1_dti[2,] <- c("Brain region", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value","B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_2tp_genr_m1_dti[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_2tp_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_2tp_genr_m1_dti[rowcount_fit, 1:5] <- c(x, lavInspect(CLPM_2tp_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm_2tp <- rowcount_clpm_2tp + 1
  rowcount_fit <- rowcount_fit + 1
}


#Reset rowcount to keep track of where we are in the loop
rowcount_clpm_2tp <- 1
rowcount_fit <- 1

#Model 2
#T1w
#Specify the model for all brain morphology measures
for(x in structs3_genr[1:43]){
  CLPM_2tp_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_9m_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
    x,"_f09_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

    #Estimate time dependent predictors
    sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Variances
    x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ', # Residual variances
    x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
```

```
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

  #Fit the model specified above
  CLPM_2tp_genr_fit <- lavaan(CLPM_2tp_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_2tp_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_2tp_genr_fit,
                                survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_2tp_genr_m2_t1w[(rowcount_clpm_2tp+2),1] <- x
  #Cross-sectional associations T1
  results_CLPM_2tp_genr_m2_t1w[c((rowcount_clpm_2tp+2)),2:4] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(21),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_2tp_genr_m2_t1w[c((rowcount_clpm_2tp+2)),5:7] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1),c(5,6,8)]
  #MRI
  results_CLPM_2tp_genr_m2_t1w[c((rowcount_clpm_2tp+2)),8:10] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_2tp_genr_m2_t1w[c((rowcount_clpm_2tp+2)),11:13] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_2tp_genr_m2_t1w[c((rowcount_clpm_2tp+2)),14:16] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_2tp_genr_m2_t1w[1,c(3,6,9,12,15)] <-
    c("Cross-sectional","CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_2tp_genr_m2_t1w[2,] <- c("Brain region", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value","B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_2tp_genr_m2_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_2tp_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_2tp_genr_m2_t1w[rowcount_fit, 1:5] <-
    c(x, lavInspect(CLPM_2tp_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm_2tp <- rowcount_clpm_2tp + 1
  rowcount_fit <- rowcount_fit + 1
}


#Reset rowcount to keep track of where we are in the loop
rowcount_clpm_2tp <- 1
rowcount_fit <- 1
```

```
#DTI
#Specify the model for all brain morphology measures
for(x in structs3_genr[44:57]){
  CLPM_2tp_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_9m_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
    x,"_f09_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

    #Estimate time dependent predictors
    sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Variances
    x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ', # Residual variances
    x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

  #Fit the model specified above
  CLPM_2tp_genr_fit <- lavaan(CLPM_2tp_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_2tp_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_2tp_genr_fit,
                                     survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_2tp_genr_m2_dti[(rowcount_clpm_2tp+2),1] <- x
  #Cross-sectional associations T1
  results_CLPM_2tp_genr_m2_dti[c((rowcount_clpm_2tp+2)),2:4] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(21),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_2tp_genr_m2_dti[c((rowcount_clpm_2tp+2)),5:7] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1),c(5,6,8)]
  #MRI
```

```
  results_CLPM_2tp_genr_m2_dti[c((rowcount_clpm_2tp+2)),8:10] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_2tp_genr_m2_dti[c((rowcount_clpm_2tp+2)),11:13] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_2tp_genr_m2_dti[c((rowcount_clpm_2tp+2)),14:16] <-
    summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_2tp_genr_m2_dti[1,c(3,6,9,12,15)] <-
    c("Cross-sectional","CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_2tp_genr_m2_dti[2,] <- c("Brain region", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value","B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_2tp_genr_m2_dti[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_2tp_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_2tp_genr_m2_dti[rowcount_fit, 1:5] <-
    c(x, lavInspect(CLPM_2tp_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm_2tp <- rowcount_clpm_2tp + 1
  rowcount_fit <- rowcount_fit + 1
}


#Reset rowcount to keep track of where we are in the loop
rowcount_clpm_2tp <- 1
rowcount_fit <- 1

#Model 3
#Specify the model for all brain morphology measures
for(x in structs3_genr[1:43]){
  CLPM_2tp_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_9m_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
    x,"_f09_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

    #Estimate time dependent predictors
    sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled \n
```

```
  # Estimate the covariance between variables at the first wave.
  sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ', # Residual variances
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
CLPM_2tp_genr_fit <- lavaan(CLPM_2tp_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_2tp_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_2tp_genr_fit,
                                   survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_CLPM_2tp_genr_m3_t1w[(rowcount_clpm_2tp+2),1] <- x
#Cross-sectional associations T1
results_CLPM_2tp_genr_m3_t1w[c((rowcount_clpm_2tp+2)),2:4] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(23),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_2tp_genr_m3_t1w[c((rowcount_clpm_2tp+2)),5:7] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1),c(5,6,8)]
#MRI
results_CLPM_2tp_genr_m3_t1w[c((rowcount_clpm_2tp+2)),8:10] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4),c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_CLPM_2tp_genr_m3_t1w[c((rowcount_clpm_2tp+2)),11:13] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3),c(5,6,8)]
#MRI -> CBCL
results_CLPM_2tp_genr_m3_t1w[c((rowcount_clpm_2tp+2)),14:16] <-
  summary(CLPM_2tp_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2),c(5,6,8)]
#Which measure (actually only needs to be specified once)
results_CLPM_2tp_genr_m3_t1w[1,c(3,6,9,12,15)] <-
  c("Cross-sectional","CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
results_CLPM_2tp_genr_m3_t1w[2,] <- c("Brain region", "B", "S.E.", "p-value",
                            "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                            "B", "S.E.", "p-value","B", "S.E.", "p-value")
#Store summary statistics
fitmeaures_CLPM_2tp_genr_m3_t1w[rowcount_fit, 1:5] <-
  c(x, summary(CLPM_2tp_genr_CSA, fit.measures = T,
               standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
#Store raw Rsquared
rsquared_CLPM_2tp_genr_m3_t1w[rowcount_fit, 1:5] <-
```

```
    c(x, lavInspect(CLPM_2tp_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm_2tp <- rowcount_clpm_2tp + 1
  rowcount_fit <- rowcount_fit + 1
}



#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_2tp_genr_m1_t1w, "results_CLPM_2tp_genr_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_2tp_genr_m2_t1w, "results_CLPM_2tp_genr_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_2tp_genr_m3_t1w, "results_CLPM_2tp_genr_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_2tp_genr_m1_t1w, "fitmeaures_CLPM_2tp_genr_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_2tp_genr_m2_t1w, "fitmeaures_CLPM_2tp_genr_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_2tp_genr_m3_t1w, "fitmeaures_CLPM_2tp_genr_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_2tp_genr_m1_t1w, "rsquared_CLPM_2tp_genr_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_2tp_genr_m2_t1w, "rsquared_CLPM_2tp_genr_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_2tp_genr_m3_t1w, "rsquared_CLPM_2tp_genr_m3_t1w.csv",
          row.names = F, quote = F)

write.csv(results_CLPM_2tp_genr_m1_dti, "results_CLPM_2tp_genr_m1_dti.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_2tp_genr_m2_dti, "results_CLPM_2tp_genr_m2_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_2tp_genr_m1_dti, "fitmeaures_CLPM_2tp_genr_m1_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_2tp_genr_m2_dti, "fitmeaures_CLPM_2tp_genr_m2_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_2tp_genr_m1_dti, "rsquared_CLPM_2tp_genr_m1_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_2tp_genr_m2_dti, "rsquared_CLPM_2tp_genr_m2_dti.csv",
          row.names = F, quote = F)
```

2.2. ABCD Study

2.2.1. Load data

```
#Read in ABCD-data
setwd("V:/medewerkers/074008 Blok, E/ABCD_data/ABCDStudyNDA_4.0Release/")

##MRI data
#T1-weighted
#Also includes covariates age at assessment and sex and ICV
t1w_abcd <- read.table("abcd_smrip10201.txt", skip = 2, header = FALSE, sep = "")
```

```
names(t1w_abcd) <- scan("abcd_smrip10201.txt", nlines = 1, what = character(), sep = "")

#DTI
dti_abcd <- read.table("abcd_dti_p101.txt", skip = 2, header = FALSE, sep = "")
names(dti_abcd) <- scan("abcd_dti_p101.txt", nlines = 1, what = character(), sep = "")

#MRI usability
mri_usable_abcd <- read.table("abcd_imgincl01.txt", skip = 2, header = FALSE, sep = "")
names(mri_usable_abcd) <- scan("abcd_imgincl01.txt", nlines = 1, what = character(), sep = "")

##CBCL data
cbcl_abcd <- read.table("abcd_cbcls01.txt", skip = 2, header = FALSE, sep = "")
names(cbcl_abcd) <- scan("abcd_cbcls01.txt", nlines = 1, what = character(), sep = "")

##Covariates
#Education
educ_abcd <- read.table("pdem02.txt", skip = 2, header = FALSE, sep = "")
names(educ_abcd) <- scan("pdem02.txt", nlines = 1, what = character(), sep = "")

#Family structure & ethnicity
sibs_eth_abcd <- read.table("acspsw03.txt", skip = 2, header = FALSE, sep = "")
names(sibs_eth_abcd) <- scan("acspsw03.txt", nlines = 1, what = character(), sep = "")

#Handedness
handedness_abcd <- read.table("abcd_ehis01.txt", skip = 2, header = FALSE, sep = "")
names(handedness_abcd) <- scan("abcd_ehis01.txt", nlines = 1, what = character(), sep = "")

#Scanner Site
site_abcd <- read.table("abcd_lt01.txt", skip = 2, header = FALSE, sep = "")
names(site_abcd) <- scan("abcd_lt01.txt", nlines = 1, what = character(), sep = "")

##Data for sensitivity analyses
#Child cognitive performance
cog_perf_abcd <- read.table("abcd_tbss01.txt", skip = 2, header = FALSE, sep = "")
names(cog_perf_abcd) <- scan("abcd_tbss01.txt", nlines = 1, what = character(), sep = "")

#Child medication use
#NOTE: This file could not be loaded in it's original format, as it contained a
#notification in 10 rows that included the " sign, the notification was:
#<span style=color:#888;">[<span style="margin:0 2px;">No results were returned</span>]</span>
#<span #style="color:#888;">[<span style="margin:0 2px;">No results were returned</span>]</span>"
#Therefore, this notification (NOT THE SUBJECT!) was removed from the following rows:
#2664, 5840, 8557, 8732, 13373, 17722, 22713, 26755, 33949, 39272
#File was not altered in any other way, but saved under the filename with suffix _eb
meduse_abcd <- read.table("medsy01_eb.txt", skip = 2, header = FALSE, sep = "")
names(meduse_abcd) <- scan("medsy01.txt", nlines = 1, what = character(), sep = "")
```

2.2.2. Select variables and merge data

```
##MRI data
#T1-weighted
t1w_sel_abcd <-
  select(t1w_abcd, c(subjectkey, interview_age, sex, smri_vol_scs_intracranialv,
```

```
                      smri_vol_scs_tplh, smri_vol_scs_tprh,
                      smri_vol_scs_caudatelh, smri_vol_scs_caudaterh,
                      smri_vol_scs_putamenlh, smri_vol_scs_putamenrh,
                      smri_vol_scs_pallidumlh, smri_vol_scs_pallidumrh,
                      smri_vol_scs_hpuslh, smri_vol_scs_hpusrh,
                      smri_vol_scs_amygdalalh, smri_vol_scs_amygdalarh,
                      smri_vol_scs_aal, smri_vol_scs_aar,
                      eventname, (starts_with("smri_vol_cdk") &
                                    (!ends_with("total") & !ends_with("totallh") &
                                       !ends_with("totalrh")))))

#DTI
dti_sel_abcd <- select(dti_abcd, c(subjectkey, eventname,
  #FA
  dmri_dtifa_fiberat_cgclh, dmri_dtifa_fiberat_cgcrh,
  dmri_dtifa_fiberat_cstlh, dmri_dtifa_fiberat_cstrh,
  dmri_dtifa_fiberat_unclh, dmri_dtifa_fiberat_uncrh,
  dmri_dtifa_fiberat_ilflh, dmri_dtifa_fiberat_ilfrh,
  dmri_dtifa_fiberat_fmaj, dmri_dtifa_fiberat_fmin,
  dmri_dtifa_fiberat_slflh, dmri_dtifa_fiberat_slfrh,
  #MD
  dmri_dtimd_fiberat_cgclh, dmri_dtimd_fiberat_cgcrh,
  dmri_dtimd_fiberat_cstlh, dmri_dtimd_fiberat_cstrh,
  dmri_dtimd_fiberat_unclh, dmri_dtimd_fiberat_uncrh,
  dmri_dtimd_fiberat_ilflh, dmri_dtimd_fiberat_ilfrh,
  dmri_dtimd_fiberat_fmaj, dmri_dtimd_fiberat_fmin,
  dmri_dtimd_fiberat_slflh, dmri_dtimd_fiberat_slfrh))

#Usability
mri_usable_sel_abcd <-
  select(mri_usable_abcd, c(subjectkey, eventname,
                            imgincl_t1w_include, imgincl_dmri_include))

##CBCL
cbcl_sel_abcd <- select(cbcl_abcd, c(subjectkey, eventname,
                           (starts_with("cbcl_scr_syn") & ends_with("_r"))))
##Covariates
#Education
educ_sel_abcd <-
  select(educ_abcd, c("subjectkey", "eventname", "demo_prnt_ed_v2"))

#Family structure & ethnicity
sibs_eth_sel_abcd <-
  select(sibs_eth_abcd, c("subjectkey", "eventname", "rel_relationship",
                      "rel_family_id", "race_ethnicity"))

#Handedness
handedness_sel_abcd <-
  select(handedness_abcd, c("subjectkey", "eventname", "ehi_y_ss_scoreb"))

#Scanner site
site_sel_abcd <-
  select(site_abcd, c("subjectkey", "eventname", "site_id_l"))
```

```
##Data for sensitivity analyses
#Child cognitive performance
cog_perf_sel_abcd <- select(cog_perf_abcd, c(subjectkey, eventname, nihtbx_totalcomp_uncorrected))

#Child medication use
meduse_sel_abcd <- select(meduse_abcd,
                          c(subjectkey, med1_rxnorm_p, med2_rxnorm_p, med3_rxnorm_p, med4_rxnorm_p,
                            med5_rxnorm_p, med6_rxnorm_p, med7_rxnorm_p, med8_rxnorm_p, med9_rxnorm_p,
                            med10_rxnorm_p, med11_rxnorm_p, med12_rxnorm_p))


#This part is adapted from earlier work by Dr. Joshua Gray
#For original script, see the link below:
#https://github.com/jgray7700/ABCD_BMI/blob/master/ABCD%20BMI%20data%20grab.txt
#Stimulants
meduse_sel_abcd$stimulant <-
  str_count(meduse_sel_abcd$med1_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med2_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med3_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med4_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med5_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med6_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med7_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med8_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med9_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med10_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med11_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo') +
  str_count(meduse_sel_abcd$med12_rxnorm_p,
            'Adderall|Methylphenidate|Dextroamphetamine|dexmethylphenidate|Vyvanse|
            Amphetamine|Concerta|Focalin|Quillivant|Ritalin|Metadate|Evekeo')
```

```
meduse_sel_abcd$stimulant[meduse_sel_abcd$stimulant>0] <- 1

#Antidepressants
meduse_sel_abcd$antidepressants <-
  str_count(meduse_sel_abcd$med1_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med2_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med3_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med4_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med5_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med6_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med7_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med8_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med9_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med10_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
```

```
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med11_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine') +
  str_count(meduse_sel_abcd$med12_rxnorm_p,
            'Mirtazapine|Imipramine|bupropion|Bupropion|Nortriptyline|Amitriptyline|
            Fluoxetine|Citalopram|Celexa|Escitalopram|Lexapro|Fluvoxamine|Paroxetine|
            Sertraline|Zoloft|atomoxetine|Strattera|duloxetine|Cymbalta|venlafaxine|
            Venlafaxine|Effexor|Desvenlafaxine|Pristiq|selegiline|Amoxapine|Imipramine')

meduse_sel_abcd$antidepressants[meduse_sel_abcd$antidepressants>0] <- 1

#Anti-anxiety
meduse_sel_abcd$antianxiety <-
  str_count(meduse_sel_abcd$med1_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med2_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med3_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med4_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med5_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med6_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med7_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med8_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med9_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med10_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med11_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan') +
  str_count(meduse_sel_abcd$med12_rxnorm_p,
            'Xanax|Klonopin|Clonazepam|Valium|Diazepam|Lorazepam|Ativan')

meduse_sel_abcd$antianxiety[meduse_sel_abcd$antianxiety>0] <- 1

#Antipsychotics
meduse_sel_abcd$antipsychotics <-
  str_count(meduse_sel_abcd$med1_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med2_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med3_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
```

```
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med4_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med5_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med6_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med7_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med8_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med9_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med10_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med11_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon') +
  str_count(meduse_sel_abcd$med12_rxnorm_p,
            'Chlorpromazine|Perphenazine|aripiprazole|Abilify|Clozapine|olanzap|Zyprexa|
            quetiapine|Seroquel|Risperidone|ziprasidone|Geodon')

meduse_sel_abcd$antipsychotics[meduse_sel_abcd$antipsychotics>0] <- 1

#Mood stabilizers
meduse_sel_abcd$moodstabilizer <-
  str_count(meduse_sel_abcd$med1_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med2_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med3_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med4_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med5_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med6_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med7_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med8_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med9_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med10_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
```

```
  str_count(meduse_sel_abcd$med11_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium') +
  str_count(meduse_sel_abcd$med12_rxnorm_p,
            'Carbamazepine|Tegretol|Divalproex|Depakote|lamotrigine|Lamictal|Lithium')

meduse_sel_abcd$moodstabilizer[meduse_sel_abcd$moodstabilizer>0] <- 1

#Code if they use any psychotropic medication
meduse_sel_abcd$psychotropic <-
  ifelse(meduse_sel_abcd$stimulant == 1 | meduse_sel_abcd$antidepressants == 1 |
           meduse_sel_abcd$antianxiety == 1 | meduse_sel_abcd$antipsychotics == 1 |
           meduse_sel_abcd$moodstabilizer == 1, 1, 0)

#Each subject is in the dataset multiple times
#As we're interested in lifetime use, we will include each subject only
#once and for that reason include the measurement that indicates the
#use of psychotropic medication if that is present
#Remove session with highest Euler number
meduse_sel_nodups_abcd <-
  meduse_sel_abcd %>%
  arrange(desc(stimulant), desc(antidepressants), desc(antianxiety), desc(antipsychotics),
           desc(moodstabilizer)) %>%
  distinct(subjectkey, .keep_all = T)

#Merge data
df1_abcd <- merge(t1w_sel_abcd, dti_sel_abcd, by = c("subjectkey", "eventname"), all = T)
df2_abcd <- merge(df1_abcd, cbcl_sel_abcd, by = c("subjectkey", "eventname"), all = T)
df3_abcd <- merge(df2_abcd, educ_sel_abcd, by = c("subjectkey", "eventname"), all = T)
df4_abcd <- merge(df3_abcd, sibs_eth_sel_abcd, by = c("subjectkey", "eventname"), all = T)
df5_abcd <- merge(df4_abcd, handedness_sel_abcd, by = c("subjectkey", "eventname"), all = T)
df6_abcd <- merge(df5_abcd, cog_perf_sel_abcd, by = c("subjectkey", "eventname"), all = T)
df7_abcd <- merge(df6_abcd, site_sel_abcd, by = c("subjectkey", "eventname"), all = T)
df8_abcd <- merge(df7_abcd, mri_usable_sel_abcd, by = c("subjectkey", "eventname"), all = T)
df_abcd <- merge(df8_abcd, meduse_sel_nodups_abcd, by = c("subjectkey"), all = T)
```

2.2.3. Select participants and time points

```
#Select only the baseline and 2 year follow-up data
#As these waves currently include MRI data
df_sel_abcd <- subset(df_abcd, eventname == "baseline_year_1_arm_1" |
                    eventname == "2_year_follow_up_y_arm_1")

#Now pivot the dataframe wider to select those
#participants with longitudinal MRI & CBCL data
df_wide_abcd <- pivot_wider(df_sel_abcd, names_from = c("eventname"),
                            values_from = c(colnames(df_sel_abcd)[c(3:131)]))

#Select those participants with usable MRI data
#by setting all invalid measures to NA
for(x in 1:nrow(df_wide_abcd)){
  #T1w
  if(df_wide_abcd[x,"imgincl_t1w_include_baseline_year_1_arm_1"] == 0 |
     is.na(df_wide_abcd[x,"imgincl_t1w_include_baseline_year_1_arm_1"]) |
```

```
    df_wide_abcd[x,"imgincl_t1w_include_2_year_follow_up_y_arm_1"] == 0 |
    is.na(df_wide_abcd[x,"imgincl_t1w_include_2_year_follow_up_y_arm_1"])){
   df_wide_abcd[x,24:189] <- NA
  }
  #DTI
  if(df_wide_abcd[x,"imgincl_dmri_include_baseline_year_1_arm_1"] == 0 |
     is.na(df_wide_abcd[x,"imgincl_dmri_include_baseline_year_1_arm_1"]) |
     df_wide_abcd[x,"imgincl_dmri_include_2_year_follow_up_y_arm_1"] == 0 |
     is.na(df_wide_abcd[x,"imgincl_dmri_include_2_year_follow_up_y_arm_1"])){
    df_wide_abcd[x,190:237] <- NA
  }
}


#Select those participants with longitudinal data
#Participants have to have T1-weighted OR DTI at 2 timepoints
#as well as CBCL data at 2 timepoints
df_sel2_abcd <- subset(df_wide_abcd, (!is.na(rowSums(df_wide_abcd[,24:189])) |
                            !is.na(rowSums(df_wide_abcd[,190:237]))) &
                  !is.na(rowSums(df_wide_abcd[,238:259])))

#Exclude columns with just NA's
df_com_abcd <- df_sel2_abcd[,-which((colSums(is.na(df_sel2_abcd)) == nrow(df_sel2_abcd)))]
```

2.2.4. Transformations

```
##MRI data
#Combine hemispheres by calculating average of left & right hemispheres
#Create vector of all structure names
#Since the structure names are equal in all MRI datasets,
#we can just create a vector of structure names based on 1 measurement occasion
#Since all structures are included in the dataset twice (once for both hemispheres),
#we only take those belonging to the left hemisphere
t1w_cols_abcd <- colnames(select(t1w_sel_abcd, (starts_with("smri_vol") & ends_with("lh")) |
                            (starts_with("smri_vol") & !ends_with("lh")) & !ends_with("rh")))
dti_cols_abcd <- colnames(select(dti_sel_abcd, (starts_with("dmri_dti") & ends_with("lh")) |
                            (starts_with("dmri_dti") & !ends_with("lh")) & !ends_with("rh")))

#Remove hemisphere suffix
t1w_structs_abcd <- stri_remove_empty(unlist(strsplit(t1w_cols_abcd, "lh")))
dti_structs_abcd <- stri_remove_empty(unlist(strsplit(dti_cols_abcd, "lh")))

#Combine aseg and aparc vectors into one vector to loop over all of them in the next step
structs_abcd <- c(t1w_structs_abcd, dti_structs_abcd)

#Remove amygdala and nucleus accumbens volumes,
#as we analyze those structures for hemisphere specific effects
#Also remove ICV, the forceps minor & minor
structs2_abcd <- structs_abcd[! structs_abcd %in% c("smri_vol_scs_amygdala",
                                  "smri_vol_scs_aal", "smri_vol_scs_aar",
                                  "smri_vol_scs_intracranialv",
                                  "dmri_dtifa_fiberat_fmaj", "dmri_dtimd_fiberat_fmaj",
                                  "dmri_dtifa_fiberat_fmin", "dmri_dtimd_fiberat_fmin")]
```

```
#Loop over all structures and average columns of the same structure in both hemispheres
#(at the same measurement occasion)
#Go over all structures
for(x in structs2_abcd){
  #Go over all columns to identify the first column (left hemisphere)
  for(y in 1:ncol(df_com_abcd)){
    #Store the column name to find the counterpart in the other (right) hemisphere
    a <- colnames(df_com_abcd[y])
    #Go over all columns to identify the second column (right hemisphere)
    for(z in 1:ncol(df_com_abcd)){
      #Store the column name to identify the same structures in each hemisphere
      b <- colnames(df_com_abcd[z])
      #If we have the same structure for both hemispheres, calculate the mean of those

      ##MRI baseline
      if(a == paste0(x,"lh_baseline_year_1_arm_1") & b == paste0(x,"rh_baseline_year_1_arm_1")){
        newcolname <- paste0(x,"_baseline_year_1_arm_1")
        df_com_abcd[,newcolname] <- (df_com_abcd[,y] + df_com_abcd[,z])/2
      }

      ##MRI follow-up
      if(a == paste0(x,"lh_2_year_follow_up_y_arm_1") & b == paste0(x,"rh_2_year_follow_up_y_arm_1")){
        newcolname <- paste0(x,"_2_year_follow_up_y_arm_1")
        df_com_abcd[,newcolname] <- (df_com_abcd[,y] + df_com_abcd[,z])/2
      }
    }
  }
}

#Now tidy up the dataset
#Remove all original hemisphere specific variables
df_tidy_abcd <-
  select(df_com_abcd,
         -c(starts_with(c(unlist(lapply(structs2_abcd, function(x)
                       paste0(x,"lh_baseline_year_1_arm_1"))),
                        unlist(lapply(structs2_abcd, function(x)
                        paste0(x,"rh_baseline_year_1_arm_1"))),
                          unlist(lapply(structs2_abcd, function(x)
                          paste0(x,"lh_2_year_follow_up_y_arm_1"))),
                          unlist(lapply(structs2_abcd, function(x)
                          paste0(x,"rh_2_year_follow_up_y_arm_1")))))))

#Normalize to mean 0 and SD 1
mri_vars_abcd <- colnames(select(df_tidy_abcd, starts_with("smri_vol") | starts_with("dmri_dti")))
for(x in mri_vars_abcd){
  newcolname <- paste0(x, "_scaled")
  df_tidy_abcd[,newcolname] <- c(scale(df_tidy_abcd[,x]))
}

##CBCL data
df_tidy_abcd$sum_dp_t1 <-
  df_tidy_abcd$cbcl_scr_syn_anxdep_r_baseline_year_1_arm_1 +
  df_tidy_abcd$cbcl_scr_syn_attention_r_baseline_year_1_arm_1 +
```

```
  df_tidy_abcd$cbcl_scr_syn_aggressive_r_baseline_year_1_arm_1
df_tidy_abcd$sum_dp_t2 <-
  df_tidy_abcd$cbcl_scr_syn_anxdep_r_2_year_follow_up_y_arm_1 +
  df_tidy_abcd$cbcl_scr_syn_attention_r_2_year_follow_up_y_arm_1 +
  df_tidy_abcd$cbcl_scr_syn_aggressive_r_2_year_follow_up_y_arm_1

#Sqrt transform & normalize to mean 0 and SD 1
cbcl_vars_abcd <- colnames(select(df_tidy_abcd,
                                  c(starts_with("cbcl_scr_syn"), "sum_dp_t1", "sum_dp_t2")))
for(x in cbcl_vars_abcd){
  newcolname <- paste0(x, "_sqrt_scaled")
  df_tidy_abcd[,newcolname] <- c(scale(sqrt(df_tidy_abcd[,x])))
}

##Covariates

#Specify sex variables as factors
df_tidy_abcd$sex_baseline_year_1_arm_1 <- as.factor(df_tidy_abcd$sex_baseline_year_1_arm_1)
df_tidy_abcd$sex_2_year_follow_up_y_arm_1 <- as.factor(df_tidy_abcd$sex_2_year_follow_up_y_arm_1)

#Maternal education
#Low = tm 6th grade
#Middle = tm 12th grade, diploma any or highschool diploma
#High = any above
df_tidy_abcd$maternal_education <-
  as.factor(ifelse(df_tidy_abcd$demo_prnt_ed_v2_baseline_year_1_arm_1 < 7, "low",
                   ifelse(df_tidy_abcd$demo_prnt_ed_v2_baseline_year_1_arm_1 > 6 &
                            df_tidy_abcd$demo_prnt_ed_v2_baseline_year_1_arm_1 < 15, "middle",
                          ifelse(df_tidy_abcd$demo_prnt_ed_v2_baseline_year_1_arm_1 >14 &
                                   df_tidy_abcd$demo_prnt_ed_v2_baseline_year_1_arm_1 < 22,
                                 "high", NA))))

#Subset ethnicity
df_tidy_abcd$ethnicity <-
  as.factor(ifelse(df_tidy_abcd$race_ethnicity_baseline_year_1_arm_1 == 1, "White",
                   ifelse(df_tidy_abcd$race_ethnicity_baseline_year_1_arm_1 == 2, "Black",
                          ifelse(df_tidy_abcd$race_ethnicity_baseline_year_1_arm_1 == 3, "Hispanic",
                                 ifelse(df_tidy_abcd$race_ethnicity_baseline_year_1_arm_1 == 4,
                                        "Asian", "Other")))))


#Handedness
df_tidy_abcd$handedness <-
  ifelse(df_tidy_abcd$ehi_y_ss_scoreb_baseline_year_1_arm_1 == 1, 1,
         ifelse(df_tidy_abcd$ehi_y_ss_scoreb_baseline_year_1_arm_1 == 2, -1, 0))


#Now tidy up the dataset
#Remove all unscaled variables
#Remove EDUCM5 & ETHNINFv2 since they have been recategorized
#Remove columns for image QC
df_final_abcd <-
  select(df_tidy_abcd,
```

```
        -c(all_of(mri_vars_abcd), all_of(cbcl_vars_abcd),
           demo_prnt_ed_v2_baseline_year_1_arm_1,
           race_ethnicity_baseline_year_1_arm_1,
           ehi_y_ss_scoreb_baseline_year_1_arm_1,
           starts_with("med"),
           starts_with("imgincl_")))


#Fix site information - this was wrong for some children as specified in the ABCD release 4.0 notes
#issues specified in release 4.0, document 3a. NDA 4.0 Changes and Known Issues
#(title demographics, subtitle incorrect site_id_l reported)
#this document is available at https://nda.nih.gov/study.html?id=1299
#change incorrect values to correct ones as specified in the guide

df_final_abcd[df_final_abcd$subjectkey == "NDAR_INV2HYAENE6",
              "site_id_l_2_year_follow_up_y_arm_1"] <-  "site08"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INV4DVGGJE9",
              "site_id_l_baseline_year_1_arm_1"] <- "site22"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INV6JF8WUYT",
              "site_id_l_baseline_year_1_arm_1"] <- "site21"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INV6WV9X2KM",
              "site_id_l_baseline_year_1_arm_1"] <- "site22"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INV6WV9X2KM",
              "site_id_l_2_year_follow_up_y_arm_1"] <- "site21"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVC7P1CVEU",
              "site_id_l_baseline_year_1_arm_1"] <- "site17"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVG19M2F39",
              "site_id_l_baseline_year_1_arm_1"] <- "site06"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVGFCRX7YW",
              "site_id_l_baseline_year_1_arm_1"] <- "site17"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVGVPPRTDN",
              "site_id_l_baseline_year_1_arm_1"] <- "site20"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVHAP0JZTR",
              "site_id_l_baseline_year_1_arm_1"] <- "site05"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVJHCBZTEX",
              "site_id_l_baseline_year_1_arm_1"] <- "site13"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVJHCBZTEX",
              "site_id_l_2_year_follow_up_y_arm_1"] <- "site13"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVLVLHRL2N",
              "site_id_l_baseline_year_1_arm_1"] <- "site21"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVNTAR3TAF",
              "site_id_l_baseline_year_1_arm_1"] <- "site17"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVR0TYK5V9",
              "site_id_l_baseline_year_1_arm_1"] <- "site22"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVRY96FYZ8",
              "site_id_l_baseline_year_1_arm_1"] <- "site05"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVT1C2GBHB",
              "site_id_l_baseline_year_1_arm_1"] <- "site19"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVT1C2GBHB",
              "site_id_l_2_year_follow_up_y_arm_1"] <- "site19"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVUB6JP787",
              "site_id_l_baseline_year_1_arm_1"] <- "site13"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVUFF64VGJ",
```

```
              "site_id_l_baseline_year_1_arm_1"] <- "site16"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVUKPZU1JW",
              "site_id_l_baseline_year_1_arm_1"] <- "site13"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVVT14CE3D",
              "site_id_l_baseline_year_1_arm_1"] <- "site22"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVWF7C1DEL",
              "site_id_l_baseline_year_1_arm_1"] <- "site09"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVXLFHB010",
              "site_id_l_baseline_year_1_arm_1"] <- "site16"
df_final_abcd[df_final_abcd$subjectkey == "NDAR_INVY92TEZW6",
              "site_id_l_baseline_year_1_arm_1"] <- "site13"

#Exclude children that do not have the same scanner site at both waves
#As these are very few kids it is difficult for the models to properly estimate
df_final_abcd$samesite <-
  ifelse(df_final_abcd$site_id_l_baseline_year_1_arm_1 ==
     df_final_abcd$site_id_l_2_year_follow_up_y_arm_1, "yes", "no")

df_final2_abcd <- select(subset(df_final_abcd, samesite == "yes"), -c(samesite))
```

2.2.5. Imputations

```
#create imputation method vector
meth_abcd <- make.method(df_final2_abcd)
#don't impute ID numbers, CBCL or MRI variables
meth_abcd[c(1:7,12:13,15:157)] <- ""
qpredR_abcd <- quickpred(df_final2_abcd)
#ID numbers, family ID, site, CBCL or MRI variables not as predictors
qpredR_abcd[,c(1:7,12:13,15:157)] <- 0
#apply default predictor matrix rules
diag(qpredR_abcd) <- 0; qpredR_abcd[which(meth_abcd == ""),] <- 0
#test run
ini_abcd <- mice(df_final2_abcd, predictorMatrix = qpredR_abcd,
                 maxit=1, m=1, printFlag=F,method = meth_abcd)

#test logged events
ini_abcd$loggedEvents

dsImp_abcd <- mice(df_final2_abcd, predictorMatrix = qpredR_abcd,
                   maxit=30, m=30, method = meth_abcd, seed = 2021)

#Save imputed dataframe
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/2.Data")
saveRDS(dsImp_abcd, "dsImp_ABCD.rds")
```

2.2.6. Cross-lagged panel model

```
##Make list of imputed datasets to feed to lavaan
implist_abcd <- lapply(seq(dsImp_abcd$m), function(im) complete(dsImp_abcd, im))

##Create dummy variables for categorical variables
for(x in seq(dsImp_abcd$m)){
  #Sex
```

```
implist_abcd[[x]]$sex <-
  ifelse(implist_abcd[[x]]$sex_baseline_year_1_arm_1 == "F", 1, 0)
#Maternal education
implist_abcd[[x]]$maternal_education_high <-
  ifelse(implist_abcd[[x]]$maternal_education == "high", 1, 0)
implist_abcd[[x]]$maternal_education_middle <-
  ifelse(implist_abcd[[x]]$maternal_education == "middle", 1, 0)
#Child national origin
implist_abcd[[x]]$child_nationalorigin_white <-
  ifelse(implist_abcd[[x]]$ethnicity == "White", 1, 0)
implist_abcd[[x]]$child_nationalorigin_black <-
  ifelse(implist_abcd[[x]]$ethnicity == "Black", 1, 0)
implist_abcd[[x]]$child_nationalorigin_hispanic <-
  ifelse(implist_abcd[[x]]$ethnicity == "Hispanic", 1, 0)
implist_abcd[[x]]$child_nationalorigin_asian <-
  ifelse(implist_abcd[[x]]$ethnicity == "Asian", 1, 0)
#Scanner site
implist_abcd[[x]]$site1 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site01", 1, 0)
implist_abcd[[x]]$site2 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site02", 1, 0)
implist_abcd[[x]]$site3 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site03", 1, 0)
implist_abcd[[x]]$site4 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site04", 1, 0)
implist_abcd[[x]]$site5 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site05", 1, 0)
implist_abcd[[x]]$site6 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site06", 1, 0)
implist_abcd[[x]]$site7 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site07", 1, 0)
implist_abcd[[x]]$site8 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site08", 1, 0)
implist_abcd[[x]]$site9 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site09", 1, 0)
implist_abcd[[x]]$site10 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site10", 1, 0)
implist_abcd[[x]]$site11 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site11", 1, 0)
implist_abcd[[x]]$site12 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site12", 1, 0)
implist_abcd[[x]]$site13 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site13", 1, 0)
implist_abcd[[x]]$site14 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site14", 1, 0)
implist_abcd[[x]]$site15 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site15", 1, 0)
implist_abcd[[x]]$site16 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site16", 1, 0)
implist_abcd[[x]]$site17 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site17", 1, 0)
implist_abcd[[x]]$site18 <-
  ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site18", 1, 0)
```

```
  implist_abcd[[x]]$site19 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site19", 1, 0)
  implist_abcd[[x]]$site20 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site20", 1, 0)
}

#Split T1-weighted and DTI sample again
#Do this so that there are no missings in each dataframe
implist_t1w_abcd <- list()
implist_dti_abcd <- list()

for(x in seq(dsImp_abcd$m)){
  implist_t1w_abcd[[x]] <- subset(implist_abcd[[x]],
                                  !is.na(rowSums(implist_abcd[[x]][,20:105])))
}
for(x in seq(dsImp_abcd$m)){
  implist_dti_abcd[[x]] <- subset(implist_abcd[[x]],
                                  !is.na(rowSums(implist_abcd[[x]][,106:133])))
}

#Create imputationlist object
data_abcd_t1w <- imputationList(implist_t1w_abcd)
data_abcd_dti <- imputationList(implist_dti_abcd)

#Create empty dataframes to store results
results_CLPM_abcd_m1_t1w <- data.frame()
fitmeaures_CLPM_abcd_m1_t1w <- data.frame()
rsquared_CLPM_abcd_m1_t1w <- data.frame()
results_CLPM_abcd_m2_t1w <- data.frame()
fitmeaures_CLPM_abcd_m2_t1w <- data.frame()
rsquared_CLPM_abcd_m2_t1w <- data.frame()
results_CLPM_abcd_m3_t1w <- data.frame()
fitmeaures_CLPM_abcd_m3_t1w <- data.frame()
rsquared_CLPM_abcd_m3_t1w <- data.frame()

results_CLPM_abcd_m1_dti <- data.frame()
fitmeaures_CLPM_abcd_m1_dti <- data.frame()
rsquared_CLPM_abcd_m1_dti <- data.frame()
results_CLPM_abcd_m2_dti <- data.frame()
fitmeaures_CLPM_abcd_m2_dti <- data.frame()
rsquared_CLPM_abcd_m2_dti <- data.frame()

#Create list of structures to loop over
structs3_abcd <-
  c(structs2_abcd[1:39], c("smri_vol_scs_amygdalalh", "smri_vol_scs_amygdalarh",
                           "smri_vol_scs_aal", "smri_vol_scs_aar"),
    structs2_abcd[40:49], c("dmri_dtifa_fiberat_fmaj", "dmri_dtimd_fiberat_fmaj",
                            "dmri_dtifa_fiberat_fmin", "dmri_dtimd_fiberat_fmin"))

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1
```

```
#Model 1
#T1-weighted
for(x in structs3_abcd[1:43]){
  CLPM_abcd <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_t2_sqrt_scaled +', x,"_2_year_follow_up_y_arm_1_scaled", ' ~
    sum_dp_t1_sqrt_scaled +', x,"_baseline_year_1_arm_1_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_t1_sqrt_scaled ~ sex \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ sex +
    site1 + site2 + site3 + site4 + site5 + site6 + site7 + site 8 + site9 + site10 +
    site11 + site12 + site13 + site14 + site15 + site16 + site17 + site18 + site19 +
    site20
    \n \n

    #Estimate time dependent predictors
    sum_dp_t1_sqrt_scaled ~ interview_age_baseline_year_1_arm_1 \n
    sum_dp_t2_sqrt_scaled ~ interview_age_2_year_follow_up_y_arm_1 \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ interview_age_baseline_year_1_arm_1 + handedness \n ',
    x,"_2_year_follow_up_y_arm_1_scaled", ' ~ interview_age_2_year_follow_up_y_arm_1 + handedness \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_t1_sqrt_scaled ~~ ', x,"_baseline_year_1_arm_1_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_t2_sqrt_scaled ~~ ', x,"_2_year_follow_up_y_arm_1_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_t1_sqrt_scaled ~~ sum_dp_t1_sqrt_scaled \n ', # Variances
    x,"_baseline_year_1_arm_1_scaled", ' ~~ ', x,"_baseline_year_1_arm_1_scaled",
    ' \n sum_dp_t2_sqrt_scaled ~~ sum_dp_t2_sqrt_scaled \n ', # Residual variances
    x,"_2_year_follow_up_y_arm_1_scaled", ' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_abcd = svydesign(id = ~rel_family_id_baseline_year_1_arm_1, data = data_abcd_t1w)

  #Fit the model specified above
  CLPM_abcd_fit <- lavaan(CLPM_abcd, data = implist_t1w_abcd[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_abcd_CSA <- lavaan.survey(lavaan.fit = CLPM_abcd_fit,
                                 survey.design = survey_design_abcd)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_abcd_m1_t1w[(rowcount_clpm+2),1] <- x
  #Cross-sectional associations T1
  results_CLPM_abcd_m1_t1w[c((rowcount_clpm+2)),2:4] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][c(34),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_abcd_m1_t1w[c((rowcount_clpm+2)),5:7] <-
```

```
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][1,c(5,6,8)]
  #MRI
  results_CLPM_abcd_m1_t1w[c((rowcount_clpm+2)),8:10] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][4,c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_abcd_m1_t1w[c((rowcount_clpm+2)),11:13] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][3,c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_abcd_m1_t1w[c((rowcount_clpm+2)),14:16] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][2,c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_abcd_m1_t1w[1,c(3,6,9,12,15)] <-
     c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_abcd_m1_t1w[2,] <- c("Brain region", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_abcd_m1_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_abcd_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_abcd_m1_t1w[rowcount_fit, 1:5] <- c(x, lavInspect(CLPM_abcd_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 1
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_abcd_m1_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m1_t1w[,3])-
    mean(as.numeric(rsquared_CLPM_abcd_m1_t1w[,3])))
  rsquared_CLPM_abcd_m1_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m1_t1w[,5])-
    mean(as.numeric(rsquared_CLPM_abcd_m1_t1w[,5])))


#Reset rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#DTI
for(x in structs3_abcd[44:57]){
  CLPM_abcd <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_t2_sqrt_scaled +', x,"_2_year_follow_up_y_arm_1_scaled", ' ~
    sum_dp_t1_sqrt_scaled +', x,"_baseline_year_1_arm_1_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_t1_sqrt_scaled ~ sex \n ',
```

```
  x,"_baseline_year_1_arm_1_scaled", ' ~ sex +
  site1 + site2 + site3 + site4 + site5 + site6 + site7 + site 8 + site9 + site10 +
  site11 + site12 + site13 + site14 + site15 + site16 + site17 + site18 + site19 +
  site20
  \n \n

  #Estimate time dependent predictors
  sum_dp_t1_sqrt_scaled ~ interview_age_baseline_year_1_arm_1 \n
  sum_dp_t2_sqrt_scaled ~ interview_age_2_year_follow_up_y_arm_1 \n ',
  x,"_baseline_year_1_arm_1_scaled", ' ~ interview_age_baseline_year_1_arm_1 + handedness \n ',
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~ interview_age_2_year_follow_up_y_arm_1 + handedness \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_t1_sqrt_scaled ~~ ', x,"_baseline_year_1_arm_1_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_t2_sqrt_scaled ~~ ', x,"_2_year_follow_up_y_arm_1_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_t1_sqrt_scaled ~~ sum_dp_t1_sqrt_scaled \n ', # Variances
  x,"_baseline_year_1_arm_1_scaled", ' ~~ ', x,"_baseline_year_1_arm_1_scaled",
  ' \n sum_dp_t2_sqrt_scaled ~~ sum_dp_t2_sqrt_scaled \n ', # Residual variances
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_abcd = svydesign(id = ~rel_family_id_baseline_year_1_arm_1, data = data_abcd_dti)

#Fit the model specified above
CLPM_abcd_fit <- lavaan(CLPM_abcd, data = implist_dti_abcd[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_abcd_CSA <- lavaan.survey(lavaan.fit = CLPM_abcd_fit,
                               survey.design = survey_design_abcd)
#Store coefficients of interest
#Which brain region
results_CLPM_abcd_m1_dti[(rowcount_clpm+2),1] <- x
#Cross-sectional associations T1
results_CLPM_abcd_m1_dti[c((rowcount_clpm+2)),2:4] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][c(34),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_abcd_m1_dti[c((rowcount_clpm+2)),5:7] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][1,c(5,6,8)]
#MRI
results_CLPM_abcd_m1_dti[c((rowcount_clpm+2)),8:10] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][4,c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_CLPM_abcd_m1_dti[c((rowcount_clpm+2)),11:13] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][3,c(5,6,8)]
#MRI -> CBCL
results_CLPM_abcd_m1_dti[c((rowcount_clpm+2)),14:16] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][2,c(5,6,8)]
```

```
  #Which measure (actually only needs to be specified once)
  results_CLPM_abcd_m1_dti[1,c(3,6,9,12,15)] <-
     c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_abcd_m1_dti[2,] <- c("Brain region", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_abcd_m1_dti[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_abcd_CSA, fit.measures = T,
         standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_abcd_m1_dti[rowcount_fit, 1:5] <- c(x, lavInspect(CLPM_abcd_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 1
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_abcd_m1_dti[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m1_dti[,3])-
    mean(as.numeric(rsquared_CLPM_abcd_m1_dti[,3])))
  rsquared_CLPM_abcd_m1_dti[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m1_dti[,5])-
    mean(as.numeric(rsquared_CLPM_abcd_m1_dti[,5])))

#Reset rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#Model 2
#T1-weighted
for(x in structs3_abcd[1:43]){
  CLPM_abcd <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_t2_sqrt_scaled +', x,"_2_year_follow_up_y_arm_1_scaled", ' ~
    sum_dp_t1_sqrt_scaled +', x,"_baseline_year_1_arm_1_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_t1_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian +
    site1 + site2 + site3 + site4 + site5 + site6 + site7 + site 8 + site9 + site10 +
    site11 + site12 + site13 + site14 + site15 + site16 + site17 + site18 + site19 +
    site20
    \n \n

    #Estimate time dependent predictors
```

```
  sum_dp_t1_sqrt_scaled ~ interview_age_baseline_year_1_arm_1 \n
  sum_dp_t2_sqrt_scaled ~ interview_age_2_year_follow_up_y_arm_1 \n ',
  x,"_baseline_year_1_arm_1_scaled", ' ~ interview_age_baseline_year_1_arm_1 + handedness \n ',
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~ interview_age_2_year_follow_up_y_arm_1 + handedness \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_t1_sqrt_scaled ~~ ', x,"_baseline_year_1_arm_1_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_t2_sqrt_scaled ~~ ', x,"_2_year_follow_up_y_arm_1_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_t1_sqrt_scaled ~~ sum_dp_t1_sqrt_scaled \n ', # Variances
  x,"_baseline_year_1_arm_1_scaled", ' ~~ ', x,"_baseline_year_1_arm_1_scaled",
  ' \n sum_dp_t2_sqrt_scaled ~~ sum_dp_t2_sqrt_scaled \n ', # Residual variances
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_abcd = svydesign(id = ~rel_family_id_baseline_year_1_arm_1, data = data_abcd_t1w)

#Fit the model specified above
CLPM_abcd_fit <- lavaan(CLPM_abcd, data = implist_t1w_abcd[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_abcd_CSA <- lavaan.survey(lavaan.fit = CLPM_abcd_fit,
                               survey.design = survey_design_abcd)
#Store coefficients of interest
#Which brain region
results_CLPM_abcd_m2_t1w[(rowcount_clpm+2),1] <- x
#Cross-sectional associations T1
results_CLPM_abcd_m2_t1w[c((rowcount_clpm+2)),2:4] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][c(46),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_abcd_m2_t1w[c((rowcount_clpm+2)),5:7] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][1,c(5,6,8)]
#MRI
results_CLPM_abcd_m2_t1w[c((rowcount_clpm+2)),8:10] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][4,c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_CLPM_abcd_m2_t1w[c((rowcount_clpm+2)),11:13] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][3,c(5,6,8)]
#MRI -> CBCL
results_CLPM_abcd_m2_t1w[c((rowcount_clpm+2)),14:16] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][2,c(5,6,8)]
#Which measure (actually only needs to be specified once)
results_CLPM_abcd_m2_t1w[1,c(3,6,9,12,15)] <-
   c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
results_CLPM_abcd_m2_t1w[2,] <- c("Brain region", "B", "S.E.", "p-value",
                          "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                          "B", "S.E.", "p-value", "B", "S.E.", "p-value")
#Store summary statistics
```

```
  fitmeaures_CLPM_abcd_m2_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_abcd_CSA, fit.measures = T,
         standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_abcd_m2_t1w[rowcount_fit, 1:5] <- c(x, lavInspect(CLPM_abcd_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 1
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_abcd_m2_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m2_t1w[,3])-
                 mean(as.numeric(rsquared_CLPM_abcd_m2_t1w[,3])))
  rsquared_CLPM_abcd_m2_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m2_t1w[,5])-
                 mean(as.numeric(rsquared_CLPM_abcd_m2_t1w[,5])))


#Reset rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#DTI
for(x in structs3_abcd[44:57]){
  CLPM_abcd <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_t2_sqrt_scaled +', x,"_2_year_follow_up_y_arm_1_scaled", ' ~
    sum_dp_t1_sqrt_scaled +', x,"_baseline_year_1_arm_1_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_t1_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian +
    site1 + site2 + site3 + site4 + site5 + site6 + site7 + site 8 + site9 + site10 +
    site11 + site12 + site13 + site14 + site15 + site16 + site17 + site18 + site19 +
    site20
    \n \n

    #Estimate time dependent predictors
    sum_dp_t1_sqrt_scaled ~ interview_age_baseline_year_1_arm_1 \n
    sum_dp_t2_sqrt_scaled ~ interview_age_2_year_follow_up_y_arm_1 \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ interview_age_baseline_year_1_arm_1 + handedness \n ',
    x,"_2_year_follow_up_y_arm_1_scaled", ' ~ interview_age_2_year_follow_up_y_arm_1 + handedness \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_t1_sqrt_scaled ~~ ', x,"_baseline_year_1_arm_1_scaled", # Covariance
```

```
  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_t2_sqrt_scaled ~~ ', x,"_2_year_follow_up_y_arm_1_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_t1_sqrt_scaled ~~ sum_dp_t1_sqrt_scaled \n ', # Variances
  x,"_baseline_year_1_arm_1_scaled", ' ~~ ', x,"_baseline_year_1_arm_1_scaled",
  ' \n sum_dp_t2_sqrt_scaled ~~ sum_dp_t2_sqrt_scaled \n ', # Residual variances
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_abcd = svydesign(id = ~rel_family_id_baseline_year_1_arm_1, data = data_abcd_dti)

#Fit the model specified above
CLPM_abcd_fit <- lavaan(CLPM_abcd, data = implist_dti_abcd[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_abcd_CSA <- lavaan.survey(lavaan.fit = CLPM_abcd_fit,
                               survey.design = survey_design_abcd)
#Store coefficients of interest
#Which brain region
results_CLPM_abcd_m2_dti[(rowcount_clpm+2),1] <- x
#Cross-sectional associations T1
results_CLPM_abcd_m2_dti[c((rowcount_clpm+2)),2:4] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][c(46),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_abcd_m2_dti[c((rowcount_clpm+2)),5:7] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][1,c(5,6,8)]
#MRI
results_CLPM_abcd_m2_dti[c((rowcount_clpm+2)),8:10] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][4,c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_CLPM_abcd_m2_dti[c((rowcount_clpm+2)),11:13] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][3,c(5,6,8)]
#MRI -> CBCL
results_CLPM_abcd_m2_dti[c((rowcount_clpm+2)),14:16] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][2,c(5,6,8)]
#Which measure (actually only needs to be specified once)
results_CLPM_abcd_m2_dti[1,c(3,6,9,12,15)] <-
   c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
results_CLPM_abcd_m2_dti[2,] <- c("Brain region", "B", "S.E.", "p-value",
                          "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                          "B", "S.E.", "p-value", "B", "S.E.", "p-value")
#Store summary statistics
fitmeaures_CLPM_abcd_m2_dti[rowcount_fit, 1:5] <-
  c(x, summary(CLPM_abcd_CSA, fit.measures = T,
       standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
#Store rsquared
rsquared_CLPM_abcd_m2_dti[rowcount_fit, 1:5] <- c(x, lavInspect(CLPM_abcd_CSA, "rsquare"))
#Adapt rowcounts to make sure results are stored properly
rowcount_clpm <- rowcount_clpm + 1
rowcount_fit <- rowcount_fit + 1
```

```
}

#Calculate relative Rsquared
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_abcd_m2_dti[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m2_dti[,3])-
                 mean(as.numeric(rsquared_CLPM_abcd_m2_dti[,3])))
  rsquared_CLPM_abcd_m2_dti[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m2_dti[,5])-
                 mean(as.numeric(rsquared_CLPM_abcd_m2_dti[,5])))

#Reset rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#Model 3
#Specify the model for all brain morphology measures
#Only correct volumes for ICV, not DTI measures
#T1-weighted
for(x in structs3_abcd[1:43]){
  CLPM_abcd <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_t2_sqrt_scaled +', x,"_2_year_follow_up_y_arm_1_scaled", ' ~
    sum_dp_t1_sqrt_scaled +', x,"_baseline_year_1_arm_1_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_t1_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian +
    site1 + site2 + site3 + site4 + site5 + site6 + site7 + site 8 + site9 + site10 +
    site11 + site12 + site13 + site14 + site15 + site16 + site17 + site18 + site19 +
    site20
    \n \n

    #Estimate time dependent predictors
    sum_dp_t1_sqrt_scaled ~ interview_age_baseline_year_1_arm_1 \n
    sum_dp_t2_sqrt_scaled ~ interview_age_2_year_follow_up_y_arm_1 \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ interview_age_baseline_year_1_arm_1 + handedness +
    smri_vol_scs_intracranialv_baseline_year_1_arm_1_scaled \n ',
    x,"_2_year_follow_up_y_arm_1_scaled", ' ~ interview_age_2_year_follow_up_y_arm_1 + handedness +
    smri_vol_scs_intracranialv_2_year_follow_up_y_arm_1_scaled \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_t1_sqrt_scaled ~~ ', x,"_baseline_year_1_arm_1_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_t2_sqrt_scaled ~~ ', x,"_2_year_follow_up_y_arm_1_scaled",
```

```
    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_t1_sqrt_scaled ~~ sum_dp_t1_sqrt_scaled \n ', # Variances
    x,"_baseline_year_1_arm_1_scaled", ' ~~ ', x,"_baseline_year_1_arm_1_scaled",
    ' \n sum_dp_t2_sqrt_scaled ~~ sum_dp_t2_sqrt_scaled \n ', # Residual variances
    x,"_2_year_follow_up_y_arm_1_scaled", ' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_abcd = svydesign(id = ~rel_family_id_baseline_year_1_arm_1, data = data_abcd_t1w)

  #Fit the model specified above
  CLPM_abcd_fit <- lavaan(CLPM_abcd, data = implist_t1w_abcd[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_abcd_CSA <- lavaan.survey(lavaan.fit = CLPM_abcd_fit,
                                 survey.design = survey_design_abcd)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_abcd_m3_t1w[(rowcount_clpm+2),1] <- x
  #Cross-sectional associations T1
  results_CLPM_abcd_m3_t1w[c((rowcount_clpm+2)),2:4] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][c(48),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_abcd_m3_t1w[c((rowcount_clpm+2)),5:7] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][1,c(5,6,8)]
  #MRI
  results_CLPM_abcd_m3_t1w[c((rowcount_clpm+2)),8:10] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][4,c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_abcd_m3_t1w[c((rowcount_clpm+2)),11:13] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][3,c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_abcd_m3_t1w[c((rowcount_clpm+2)),14:16] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][2,c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_abcd_m3_t1w[1,c(3,6,9,12,15)] <-
     c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_abcd_m3_t1w[2,] <- c("Brain region", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_abcd_m3_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_abcd_CSA, fit.measures = T,
         standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_abcd_m3_t1w[rowcount_fit, 1:5] <- c(x, lavInspect(CLPM_abcd_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 1
  rowcount_fit <- rowcount_fit + 1
}

#Calculate relative Rsquared
```

```
  #Subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_abcd_m3_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m3_t1w[,3])-
    mean(as.numeric(rsquared_CLPM_abcd_m3_t1w[,3])))
  rsquared_CLPM_abcd_m3_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_m3_t1w[,5])-
    mean(as.numeric(rsquared_CLPM_abcd_m3_t1w[,5])))

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_abcd_m1_t1w, "results_CLPM_abcd_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_abcd_m2_t1w, "results_CLPM_abcd_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_abcd_m3_t1w, "results_CLPM_abcd_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_abcd_m1_t1w, "fitmeaures_CLPM_abcd_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_abcd_m2_t1w, "fitmeaures_CLPM_abcd_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_abcd_m3_t1w, "fitmeaures_CLPM_abcd_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_abcd_m1_t1w, "rsquared_CLPM_abcd_m1_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_abcd_m2_t1w, "rsquared_CLPM_abcd_m2_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_abcd_m3_t1w, "rsquared_CLPM_abcd_m3_t1w.csv",
          row.names = F, quote = F)

write.csv(results_CLPM_abcd_m1_dti, "results_CLPM_abcd_m1_dti.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_abcd_m2_dti, "results_CLPM_abcd_m2_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_abcd_m1_dti, "fitmeaures_CLPM_abcd_m1_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_abcd_m2_dti, "fitmeaures_CLPM_abcd_m2_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_abcd_m1_dti, "rsquared_CLPM_abcd_m1_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_abcd_m2_dti, "rsquared_CLPM_abcd_m2_dti.csv",
          row.names = F, quote = F)
```

2.2.7. False Discovery Rate correction (Benjamini-Hochberg)

```
#Subset results into hypothesis driven and exploratory
results_CLPM_abcd_m3_hypothesisdriven <-
  rbind(results_CLPM_abcd_m3_t1w[c(20,18,42:45),], results_CLPM_abcd_m2_dti[c(3,5,15,8,10,16),])
results_CLPM_abcd_m3_exploratory <-
  rbind(results_CLPM_abcd_m3_t1w[c(3:17,19,21:41),], results_CLPM_abcd_m2_dti[c(4,6:7,9,11:14),])

#Create empty FDR dataframe
fdr_hypothesisdriven  <- matrix(nrow = nrow(results_CLPM_abcd_m3_hypothesisdriven)*5, ncol = 3)
```

```
fdr_exploratory  <- matrix(nrow = nrow(results_CLPM_abcd_m3_exploratory)*5, ncol = 3)

#Bind all pvalues together
pvals_all_hypothesisdriven <- c(results_CLPM_abcd_m3_hypothesisdriven[,4],
                                results_CLPM_abcd_m3_hypothesisdriven[,7],
                                results_CLPM_abcd_m3_hypothesisdriven[,10],
                                results_CLPM_abcd_m3_hypothesisdriven[,13],
                                results_CLPM_abcd_m3_hypothesisdriven[,16])
pvals_all_exploratory <- c(results_CLPM_abcd_m3_exploratory[,4],
                           results_CLPM_abcd_m3_exploratory[,7],
                           results_CLPM_abcd_m3_exploratory[,10],
                           results_CLPM_abcd_m3_exploratory[,13],
                           results_CLPM_abcd_m3_exploratory[,16])

#Calculate threshold for each test
for(x in 1:nrow(fdr_hypothesisdriven)){
  thresh_temp <- x/nrow(fdr_hypothesisdriven)*0.05
  if(x == 1){
    thresholds_hypothesisdriven <- thresh_temp
  } else {
    thresholds_hypothesisdriven <- c(thresholds_hypothesisdriven, thresh_temp)
  }
}

for(x in 1:nrow(fdr_exploratory)){
  thresh_temp <- x/nrow(fdr_exploratory)*0.05
  if(x == 1){
    thresholds_exploratory <- thresh_temp
  } else {
    thresholds_exploratory <- c(thresholds_exploratory, thresh_temp)
  }
}

#Fill FDR dataframes
fdr_hypothesisdriven[,1] <- sort(as.numeric(pvals_all_hypothesisdriven))
fdr_hypothesisdriven[,2] <- thresholds_hypothesisdriven
fdr_hypothesisdriven[,3] <-
    ifelse(as.numeric(fdr_hypothesisdriven[,1]) <
             as.numeric(fdr_hypothesisdriven[,2]), "sig", "nonsig")

fdr_exploratory[,1] <- sort(as.numeric(pvals_all_exploratory))
fdr_exploratory[,2] <- thresholds_exploratory
fdr_exploratory[,3] <-
    ifelse(as.numeric(fdr_exploratory[,1]) <
             as.numeric(fdr_exploratory[,2]), "sig", "nonsig")

#Add correction to tables
#Store significance after FDR-BH as * in
results_CLPM_abcd_m3_hypothesisdriven[,"sig_CS"] <-
  ifelse(as.numeric(results_CLPM_abcd_m3_hypothesisdriven[,4]) <= 0.025, "*", "")
results_CLPM_abcd_m3_hypothesisdriven[,"sig_AR_CBCL"] <-
  ifelse(as.numeric(results_CLPM_abcd_m3_hypothesisdriven[,7]) <= 0.025, "*", "")
results_CLPM_abcd_m3_hypothesisdriven[,"sig_AR_MRI"] <-
```

Table 5: Results hypothesis driven CLPM ABCD

	Brain region	CS B	SE	p		AR CBCL B	SE	p		AR MRI B	SE	p		CL CBCL>MRI B	SE	p		CL MRI>CBCL B	SE	p
20	smri_vol_cdk_mobfr	0.00	0.00	0.24		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.01	0.57		0.00	0.01	0.58
18	smri_vol_cdk_lobfr	0.00	0.00	0.27		0.71	0.01	0	*	0.01	0	0.01	*	0.01	0.00	0.23		0.00	0.01	0.62
42	smri_vol_scs_amygdalalh	0.00	0.00	0.57		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.01	0.63		-0.01	0.01	0.45
43	smri_vol_scs_amygdalarh	0.00	0.00	0.52		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.01	0.69		0.00	0.01	0.77
44	smri_vol_scs_aal	0.00	0.01	0.58		0.71	0.01	0	*	0.01	0	-0.02	*	-0.02	0.01	0.01	*	0.01	0.01	0.28
45	smri_vol_scs_aar	-0.01	0.00	0.06		0.71	0.01	0	*	0.01	0	-0.02	*	-0.02	0.01	0.00	*	0.01	0.01	0.30
3	dmri_dtifa_fiberat_cgc	-0.01	0.01	0.06		0.70	0.01	0	*	0.03	0	0.00	*	0.00	0.01	0.94		0.02	0.01	0.03
5	dmri_dtifa_fiberat_unc	0.00	0.01	0.85		0.70	0.01	0	*	0.06	0	0.01	*	0.01	0.01	0.24		0.01	0.01	0.21
15	dmri_dtifa_fiberat_fmin	-0.01	0.01	0.22		0.70	0.01	0	*	0.03	0	0.00	*	0.00	0.01	0.71		0.00	0.01	0.98
8	dmri_dtimd_fiberat_cgc	-0.01	0.01	0.15		0.70	0.01	0	*	0.08	0	-0.01	*	-0.01	0.01	0.31		-0.02	0.01	0.14
10	dmri_dtimd_fiberat_unc	-0.01	0.01	0.11		0.70	0.01	0	*	0.11	0	-0.02	*	-0.02	0.01	0.08		0.00	0.01	0.66
16	dmri_dtimd_fiberat_fmin	0.00	0.01	0.90		0.70	0.01	0	*	0.04	0	-0.02	*	-0.02	0.01	0.15		0.01	0.01	0.44

```
  ifelse(as.numeric(results_CLPM_abcd_m3_hypothesisdriven[,10]) <= 0.025, "*", "")
results_CLPM_abcd_m3_hypothesisdriven[,"sig_CL_CBCLMRI"] <-
  ifelse(as.numeric(results_CLPM_abcd_m3_hypothesisdriven[,13]) <= 0.025, "*", "")
results_CLPM_abcd_m3_hypothesisdriven[,"sig_CL_MRICBCL"] <-
  ifelse(as.numeric(results_CLPM_abcd_m3_hypothesisdriven[,16]) <= 0.025, "*", "")

results_CLPM_abcd_m3_exploratory[,"sig_CS"] <-
  ifelse(as.numeric(results_CLPM_abcd_m3_exploratory[,4]) <= 0.024, "*", "")
results_CLPM_abcd_m3_exploratory[,"sig_AR_CBCL"] <-
  ifelse(as.numeric(results_CLPM_abcd_m3_exploratory[,7]) <= 0.024, "*", "")
results_CLPM_abcd_m3_exploratory[,"sig_AR_MRI"] <-
  ifelse(as.numeric(results_CLPM_abcd_m3_exploratory[,10]) <= 0.024, "*", "")
results_CLPM_abcd_m3_exploratory[,"sig_CL_CBCLMRI"] <-
  ifelse(as.numeric(results_CLPM_abcd_m3_exploratory[,13]) <= 0.024, "*", "")
results_CLPM_abcd_m3_exploratory[,"sig_CL_MRICBCL"] <-
  ifelse(as.numeric(results_CLPM_abcd_m3_exploratory[,16]) <= 0.024, "*", "")

#Print tables
#Hypothesis driven
results_CLPM_abcd_m3_hypothesisdriven[,c(2:16)] <-
  sapply(results_CLPM_abcd_m3_hypothesisdriven[,c(2:16)], as.numeric)
results_CLPM_abcd_m3_hypothesisdriven[,c(1:4,17,5:7,18,9:11,19,11:13,20,14:16,21)] %>%
  kable(digits = 2,
        format = "latex",
      caption="Results hypothesis driven CLPM ABCD",
      col.names = c("Brain region",
                    "CS B", "SE", "p", "",
                    "AR CBCL B", "SE", "p", "",
                    "AR MRI B", "SE", "p", "",
                    "CL CBCL>MRI B", "SE", "p", "",
                    "CL MRI>CBCL B", "SE", "p", ""),
      align="r") %>%
  kable_classic(full_width = F, html_font = "helvetica")
```

```
#Exploratory
results_CLPM_abcd_m3_exploratory[,c(2:16)] <-
  sapply(results_CLPM_abcd_m3_exploratory[,c(2:16)], as.numeric)
results_CLPM_abcd_m3_exploratory[,c(1:4,17,5:7,18,9:11,19,11:13,20,14:16,21)] %>%
  kable(digits = 2,
        format = "latex",
      caption="Results exploratory CLPM ABCD",
      col.names = c("Brain region",
```

```
                    "CS B", "SE", "p", "",
                    "AR CBCL B", "SE", "p", "",
                    "AR MRI B", "SE", "p", "",
                    "CL CBCL>MRI B", "SE", "p", "",
                    "CL MRI>CBCL B", "SE", "p", ""),
      align="r") %>%
  kable_classic(full_width = F, html_font = "helvetica")
```

Table 6: Results exploratory CLPM ABCD

	Brain region	CS B	SE	p		AR CBCL B	SE	p		AR MRI B	SE	p		CL CBCL>MRI B	SE	p		CL MRI>CBCL B	SE	p
3	smri_vol_scs_tp	0.00	0.00	0.65		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.59		-0.01	0.01	0.57
4	smri_vol_scs_caudate	0.00	0.00	0.05		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.78		0.00	0.01	0.91
5	smri_vol_scs_putamen	0.00	0.00	0.66		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.77		0.00	0.01	0.89
6	smri_vol_scs_pallidum	0.00	0.00	0.51		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.01	0.92		-0.01	0.01	0.25
7	smri_vol_scs_hpus	-0.01	0.00	0.00	*	0.71	0.01	0	*	0.00	0	-0.01	*	-0.01	0.00	0.04		0.01	0.01	0.46
8	smri_vol_cdk_bankssts	0.00	0.00	0.13		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.67		0.00	0.01	0.88
9	smri_vol_cdk_cdacate	0.00	0.00	0.33		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.42		-0.01	0.01	0.41
10	smri_vol_cdk_cdmdfr	0.00	0.00	0.18		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.97		0.00	0.01	0.59
11	smri_vol_cdk_cuneus	0.00	0.00	0.09		0.71	0.01	0	*	0.00	0	0.01	*	0.01	0.00	0.07		-0.01	0.01	0.50
12	smri_vol_cdk_ehinal	-0.01	0.00	0.05		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.01	0.50		0.00	0.01	0.70
13	smri_vol_cdk_fusiform	0.00	0.00	0.65		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.11		0.00	0.01	0.96
14	smri_vol_cdk_ifpl	0.00	0.00	0.24		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.08		0.00	0.01	0.83
15	smri_vol_cdk_iftm	0.00	0.00	0.45		0.71	0.01	0	*	0.00	0	0.01	*	0.01	0.00	0.08		0.01	0.01	0.46
16	smri_vol_cdk_ihcate	0.00	0.00	0.95		0.71	0.01	0	*	0.00	0	0.01	*	0.01	0.00	0.06		0.00	0.01	0.88
17	smri_vol_cdk_locc	0.00	0.00	0.10		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.44		0.00	0.01	0.84
19	smri_vol_cdk_lingual	0.00	0.00	0.10		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.57		0.01	0.01	0.37
21	smri_vol_cdk_mdtm	0.00	0.00	0.60		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.46		0.02	0.01	0.04
22	smri_vol_cdk_parahpal	0.00	0.00	0.85		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.92		0.00	0.01	0.69
23	smri_vol_cdk_paracn	0.00	0.00	0.79		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.78		0.01	0.01	0.28
24	smri_vol_cdk_parsopc	0.00	0.00	0.42		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.32		0.01	0.01	0.16
25	smri_vol_cdk_parsobis	0.00	0.00	0.70		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.86		-0.01	0.01	0.50
26	smri_vol_cdk_parstgris	0.00	0.00	0.67		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.49		0.02	0.01	0.07
27	smri_vol_cdk_pericc	0.00	0.00	0.33		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.58		0.00	0.01	0.81
28	smri_vol_cdk_postcn	0.00	0.00	0.09		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.34		0.01	0.01	0.50
29	smri_vol_cdk_ptcate	0.00	0.00	0.03		0.71	0.01	0	*	0.00	0	0.01	*	0.01	0.00	0.01	*	-0.01	0.01	0.47
30	smri_vol_cdk_precn	0.00	0.00	0.08		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.35		0.01	0.01	0.49
31	smri_vol_cdk_pc	0.00	0.00	0.83		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.53		0.01	0.01	0.35
32	smri_vol_cdk_rracate	0.00	0.00	0.90		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.50		0.00	0.01	0.96
33	smri_vol_cdk_rrmdfr	0.00	0.00	0.14		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.45		0.00	0.01	0.70
34	smri_vol_cdk_sufr	0.00	0.00	0.72		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.93		0.01	0.01	0.47
35	smri_vol_cdk_supl	0.00	0.00	0.79		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.34		0.00	0.01	0.71
36	smri_vol_cdk_sutm	0.00	0.00	0.80		0.71	0.01	0	*	0.00	0	0.00	*	0.00	0.00	0.17		0.00	0.01	0.85
37	smri_vol_cdk_sm	0.00	0.00	0.64		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.54		0.00	0.01	0.74
38	smri_vol_cdk_frpole	0.00	0.00	0.48		0.71	0.01	0	*	0.01	0	-0.01	*	-0.01	0.01	0.29		-0.01	0.01	0.25
39	smri_vol_cdk_tmpole	-0.01	0.00	0.04		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.01	0.64		0.00	0.01	0.84
40	smri_vol_cdk_trvtm	0.00	0.00	0.41		0.71	0.01	0	*	0.00	0	0.01	*	0.01	0.00	0.13		0.01	0.01	0.39
41	smri_vol_cdk_insula	0.00	0.00	0.57		0.71	0.01	0	*	0.01	0	0.00	*	0.00	0.00	0.65		0.00	0.01	0.60
42	dmri_dtifa_fiberat_cst	0.00	0.01	0.50		0.70	0.01	0	*	0.04	0	-0.01	*	-0.01	0.01	0.18		0.00	0.01	0.99
61	dmri_dtifa_fiberat_ilf	0.00	0.01	0.92		0.70	0.01	0	*	0.06	0	-0.01	*	-0.01	0.01	0.12		0.02	0.01	0.06
71	dmri_dtifa_fiberat_slf	0.00	0.01	0.50		0.70	0.01	0	*	0.06	0	0.00	*	0.00	0.01	0.84		0.01	0.01	0.54
91	dmri_dtimd_fiberat_cst	0.00	0.01	0.95		0.70	0.01	0	*	0.11	0	-0.02	*	-0.02	0.01	0.15		0.01	0.01	0.36
111	dmri_dtimd_fiberat_ilf	-0.01	0.01	0.04		0.70	0.01	0	*	0.11	0	-0.01	*	-0.01	0.01	0.07		0.00	0.01	0.73
121	dmri_dtimd_fiberat_slf	-0.01	0.01	0.06		0.70	0.01	0	*	0.12	0	-0.01	*	-0.01	0.01	0.35		-0.01	0.01	0.22
131	dmri_dtifa_fiberat_fmaj	0.00	0.01	0.63		0.70	0.01	0	*	0.02	0	-0.01	*	-0.01	0.01	0.45		0.00	0.01	0.79
141	dmri_dtimd_fiberat_fmaj	-0.01	0.01	0.09		0.70	0.01	0	*	0.04	0	-0.01	*	-0.01	0.01	0.54		0.00	0.01	0.74

```
#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_abcd_m3_hypothesisdriven, "results_CLPM_abcd_m3_hypothesisdriven.csv",
          row.names = F, quote = F)
write.csv(results_CLPM_abcd_m3_exploratory, "results_CLPM_abcd_m3_exploratory.csv",
          row.names = F, quote = F)
```

3. Table 1

```
#Create separate dataframe for each sample with demographic variables
df_demographic_genr <-
  select(df_final_genr, c(IDC, GENDER,
                          age_child_mri_f05, age_child_mri_f09, age_child_mri_f13,
                          agechild_GR1075, AgeChild_CBCL9m, AGECHILD_GR1093,
                          HC12_F5, HC12_F9, HC12_F13,
```

```
                          maternal_education, child_nationalorigin))

df_demographic_abcd <-
  select(df_final_abcd, c(subjectkey, sex_baseline_year_1_arm_1,
                          interview_age_baseline_year_1_arm_1,
                          interview_age_2_year_follow_up_y_arm_1,
                          handedness, maternal_education, ethnicity))

#Set biological sex in ABCD Study to factor instead of character
df_demographic_abcd$sex_baseline_year_1_arm_1 <-
    as.factor(df_demographic_abcd$sex_baseline_year_1_arm_1)

#Create empty matrices to fill
table1_genr <- matrix(ncol = 2, nrow = 22)
table1_abcd <- matrix(ncol = 2, nrow = 17)

#Specify distributions of continuous variables
distributions_genr <-
  matrix(ncol = 1, nrow = 9,
         dimnames = list(c("age_child_mri_f05", "age_child_mri_f09", "age_child_mri_f13",
                           "agechild_GR1075", "AgeChild_CBCL9m", "AGECHILD_GR1093",
                           "HC12_F5", "HC12_F9", "HC12_F13"),
                                    c("distribution")))
distributions_genr[,1] <- c("normal", "normal", "normal",
                        "normal", "normal", "normal",
                        "skewed", "skewed", "skewed")

distributions_abcd <-
  matrix(ncol = 1, nrow = 3,
         dimnames = list(c("interview_age_baseline_year_1_arm_1",
                           "interview_age_2_year_follow_up_y_arm_1",
                           "handedness"),
                                    c("distribution")))
distributions_abcd[,1] <- c("normal", "normal", "skewed")

#Create tables with demographics

#Specify rowcount to keep track of where we are in the loop
rowcount <- 0

#Generation R
for(x in colnames(df_demographic_genr)[2:13]){
  rowcount <- rowcount + 1
  #Calculate means & sds for continuous variables
  if(is.numeric(df_demographic_genr[[x]])){
    #Store the total n for continuous measures
    if(is.na(mean(df_demographic_genr[[x]]))){
      table1_genr[rowcount,1] <-
        nrow(df_demographic_genr) - summary(df_demographic_genr[[x]])[[7]]
    } else {
      table1_genr[rowcount,1] <- nrow(df_demographic_genr)
    }
    if(distributions_genr[x,] == "normal"){
```

```
      table1_genr[rowcount,2] <-
        paste0(round(mean(df_demographic_genr[[x]], na.rm = T),1),
               " (", round(sd(df_demographic_genr[[x]], na.rm = T),1), ")")
    }
    if(distributions_genr[x,] == "skewed"){
      table1_genr[rowcount,2] <-
        paste0(round(summary(df_demographic_genr[[x]])[[3]],1),
               " (",round(summary(df_demographic_genr[[x]])[[2]],1),
               "-", round(summary(df_demographic_genr[[x]])[[5]],1),")")
    }
  }
  if(!is.numeric(df_demographic_genr[[x]])){
    #Calculate n and % for categorical variables
    for(y in 1:length(summary(df_demographic_genr[,x]))){
      rowcount <- rowcount + 1
      table1_genr[rowcount,1] <- summary(df_demographic_genr[[x]])[[y]]
      table1_genr[rowcount,2] <-
        paste0(round(prop.table(summary(df_demographic_genr[[x]]))[[y]]*100,1),"%")
    }
  }
}

#Reset rowcount
rowcount <- 0

#ABCD Study
for(x in colnames(df_demographic_abcd)[2:7]){
  rowcount <- rowcount + 1
  #Calculate means & sds for continuous variables
  if(is.numeric(df_demographic_abcd[[x]])){
    #Store the total n for continuous measures
    if(is.na(mean(df_demographic_abcd[[x]]))){
      table1_abcd[rowcount,1] <-
        nrow(df_demographic_abcd) - summary(df_demographic_abcd[[x]])[[7]]
    } else {
      table1_abcd[rowcount,1] <- nrow(df_demographic_abcd)
    }
    if(distributions_abcd[x,] == "normal"){
      table1_abcd[rowcount,2] <-
        paste0(round(mean(df_demographic_abcd[[x]], na.rm = T),1),
               " (", round(sd(df_demographic_abcd[[x]], na.rm = T),1), ")")
    }
    if(distributions_abcd[x,] == "skewed"){
      table1_abcd[rowcount,2] <-
        paste0(round(summary(df_demographic_abcd[[x]])[[3]],2),
               " (",round(summary(df_demographic_abcd[[x]])[[2]],1),
               "-", round(summary(df_demographic_abcd[[x]])[[5]],1),")")
    }
  }
  if(!is.numeric(df_demographic_abcd[[x]])){
    #Calculate n and % for categorical variables
    for(y in 1:length(summary(df_demographic_abcd[,x]))){
      rowcount <- rowcount + 1
```

```
      table1_abcd[rowcount,1] <- summary(df_demographic_abcd[[x]])[[y]]
      table1_abcd[rowcount,2] <-
        paste0(round(prop.table(summary(df_demographic_abcd[[x]]))[[y]]*100,1),"%")
    }
  }
}

#Give comprehensible row and column names
rownames(table1_genr) <-
  c("Child biological sex (%)", levels(df_demographic_genr$GENDER),
    "Age MRI T1 (M, SD)", "Age MRI T2 (M, SD)", "Age MRI T3 (M, SD)",
    "Age CBCL T1 (M, SD)", "Age CBCL T2 (M, SD)", "Age CBCL T3 (M, SD)",
    "Handedness T1 (Median, IQR)", "Handedness T2 (Median, IQR)",
    "Handedness T3 (Median, IQR)",
    "Maternal education (%)", levels(df_demographic_genr$maternal_education), "NA",
    "Child national origin (%)", levels(df_demographic_genr$child_nationalorigin), "NA")
colnames(table1_genr) <- c("n", "(M, SD)/%/(Median, IQR)")

rownames(table1_abcd) <-
  c("Child biological sex (%)", levels(df_demographic_abcd$sex_baseline_year_1_arm_1),
    "Age T1 (M, SD)", "Age T2 (M, SD)",
    "Handedness (Median, IQR)",
    "Maternal education (%)", levels(df_demographic_abcd$maternal_education), "NA",
    "Child national origin (%)", levels(df_demographic_abcd$ethnicity))
colnames(table1_abcd) <- c("n", "(M, SD)/%/(Median, IQR)")

#Print tables
table1_genr %>%
  kable(digits = 2,
      caption="Sample charachteristics Generation R",
      col.names = c("n", "(M, SD)/%/(Median, IQR)"),
      align="r") %>%
  kable_classic(full_width = T, html_font = "helvetica")
```

Table 7: Sample charachteristics Generation R

	n	(M, SD)/%/(Median, IQR)
Child biological sex (%)	NA	NA
boy	911	49.4%
girl	932	50.6%
Age MRI T1 (M, SD)	687	7.8 (1)
Age MRI T2 (M, SD)	1706	10.1 (0.6)
Age MRI T3 (M, SD)	1717	13.9 (0.5)
Age CBCL T1 (M, SD)	1778	6 (0.4)
Age CBCL T2 (M, SD)	1730	9.7 (0.3)
Age CBCL T3 (M, SD)	1736	13.5 (0.3)
Handedness T1 (Median, IQR)	707	0.8 (0.6-0.9)
Handedness T2 (Median, IQR)	1644	0.8 (0.7-1)
Handedness T3 (Median, IQR)	1711	0.8 (0.7-1)
Maternal education (%)	NA	NA
high	1096	59.5%
low	41	2.2%
middle	631	34.2%
NA	75	4.1%
Child national origin (%)	NA	NA
Dutch	1215	65.9%
non western	448	24.3%
other western	172	9.3%
NA	8	0.4%

```
table1_abcd %>%
  kable(digits = 2,
      caption="Sample charachteristics ABCD",
      col.names = c("n", "(M, SD)/%/(Median, IQR)"),
      align="r") %>%
  kable_classic(full_width = T, html_font = "helvetica")
```

Table 8: Sample charachteristics ABCD

	n	(M, SD)/%/(Median, IQR)
Child biological sex (%)	NA	NA
F	2973	46.5%
M	3420	53.5%
Age T1 (M, SD)	6393	119 (7.4)
Age T2 (M, SD)	6393	143 (7.7)
Handedness (Median, IQR)	6393	1 (1-1)
Maternal education (%)	NA	NA
high	5423	84.8%
low	38	0.6%
middle	922	14.4%
NA	10	0.2%
Child national origin (%)	NA	NA
Asian	126	2%
Black	777	12.2%
Hispanic	1263	19.8%
Other	647	10.1%
White	3580	56%

```
#Save results
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(table1_genr, "table1_genr.csv", quote = F)
write.csv(table1_abcd, "table1_abcd.csv", quote = F)
```

4. Sensitivity Analyses

In sensitivity analysis 1-4 we will only use those regions that had significant cross-lagged paths, thus we will define these prior to going to each individual sensitivity analysis.

```
#Define which regions had significant CL paths
sig_regions_genr_clpm_t1w <- c("Hippocampus_vol", "temporalpole_vol")
sig_regions_genr_clpm_dti <- c("unc_FA", "cst_FA")
sig_regions_genr_riclpm_t1w <- c("superiorparietal_vol")
#DTI is not included for the riclpm, as there were no significant CL results
sig_regions_abcd_clpm_t1w <- c("smri_vol_scs_aal", "smri_vol_scs_aar", "smri_vol_cdk_ptcate")
#DTI is not included for the abcd study, as there were no significant CL results
```

4.1 CBCL syndrome scales underlying CBCL-DP

In the first sensitivity analysis, we re-analyze the cross-lagged panel models that showed a significant effect in the cross-lagged paths, with the three syndrome scales underlying the CBCL-DP (Anxious/Depressed, Attention Problems and Aggressive Behavior), to assess to what extent findings are driven by one or more particular syndrome scale.

4.1.1 Generation R

```
##Make list of imputed datasets to feed to lavaan
implist_genr <- lapply(seq(dsImp_genr$m), function(im) complete(dsImp_genr, im))

##Create dummy variables for categorical variables
for(x in seq(dsImp_genr$m)){
```

```
  #Sex
  implist_genr[[x]]$sex <-
    ifelse(implist_genr[[x]]$GENDER == "girl", 1, 0)
  #Maternal education
  implist_genr[[x]]$maternal_education_high <-
    ifelse(implist_genr[[x]]$maternal_education == "high", 1, 0)
  implist_genr[[x]]$maternal_education_middle <-
    ifelse(implist_genr[[x]]$maternal_education == "middle", 1, 0)
  #Child national origin
  implist_genr[[x]]$child_nationalorigin_dutch <-
    ifelse(implist_genr[[x]]$child_nationalorigin == "Dutch", 1, 0)
  implist_genr[[x]]$child_nationalorigin_wes <-
    ifelse(implist_genr[[x]]$child_nationalorigin == "other western", 1, 0)
}

#Split T1-weighted and DTI sample
#Do this so that there are no missings in each dataframe
implist_t1w_genr <- list()
implist_dti_genr <- list()

for(x in seq(dsImp_genr$m)){
  implist_t1w_genr[[x]] <- subset(implist_genr[[x]],
                                  implist_genr[[x]]$nscans > 1)
}
for(x in seq(dsImp_genr$m)){
  implist_dti_genr[[x]] <- subset(implist_genr[[x]],
                                  implist_genr[[x]]$nscans_dti > 1)
}

#Create imputationlist object
data_genr_t1w <- imputationList(implist_t1w_genr)
data_genr_dti <- imputationList(implist_dti_genr)

#Define CBCL syndrome scales
#Suffixes will be added within CLPM
cbcl_scales_genr <- c("sum_anx", "sum_att", "sum_agg")

#Create empty dataframes to store results
results_CLPM_genr_sens1_m3_t1w <- data.frame()
fitmeaures_CLPM_genr_sens1_m3_t1w <- data.frame()
rsquared_CLPM_genr_sens1_m3_t1w <- data.frame()

results_CLPM_genr_sens1_m2_dti <- data.frame()
fitmeaures_CLPM_genr_sens1_m2_dti <- data.frame()
rsquared_CLPM_genr_sens1_m2_dti <- data.frame()

results_RICLPM_genr_sens1_m3_t1w <- data.frame()
fitmeaures_RICLPM_genr_sens1_m3_t1w <- data.frame()
rsquared_RICLPM_genr_sens1_m3_t1w <- data.frame()

#CLPM
#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
```

```
rowcount_fit <- 1

#T1w
#Specify the model for all brain morphology measures
for(a in cbcl_scales_genr){
  for(x in sig_regions_genr_clpm_t1w){
    CLPM_genr <- paste0(
      ' \n
  # Estimate the lagged effects between the variables
    ', a, "_9m_sqrt_scaled",' +', x,"_f09_scaled", ' ~
    ', a, "_5_sqrt_scaled",' +', x,"_f05_scaled",
      ' \n ' , a, "_14_sqrt_scaled",' +', x,"_f13_scaled", ' ~
    ', a, "_9m_sqrt_scaled",' +', x,"_f09_scaled",

      #Estimate time independent predictors
      ' \n \n', a, "_5_sqrt_scaled",' ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
      x,"_f05_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

    #Estimate time dependent predictors
    ', a, "_5_sqrt_scaled",' ~ agechild_GR1075 \n
    ', a, "_9m_sqrt_scaled",' ~ AgeChild_CBCL9m \n
    ', a, "_14_sqrt_scaled",' ~ AGECHILD_GR1093 \n ',
      x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled \n ',
      x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled \n ',
      x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled \n

    # Estimate the covariance between variables at the first wave.
    ', a, "_5_sqrt_scaled",' ~~ ', x,"_f05_scaled", # Covariance

      # Estimate the covariances between the residuals of variables
      ' \n ', a, "_9m_sqrt_scaled",' ~~ ', x,"_f09_scaled",
      ' \n ', a, "_14_sqrt_scaled",' ~~ ', x,"_f13_scaled",

      # Estimate the (residual) variance of variables of interest
      ' \n ', a, "_5_sqrt_scaled",' ~~ ', a, "_5_sqrt_scaled",' \n ', # Variances
      x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
      ' \n ', a, "_9m_sqrt_scaled",' ~~ ', a, "_9m_sqrt_scaled",' \n ', # Residual variances
      x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
      ' \n ', a, "_14_sqrt_scaled",' ~~ ', a, "_14_sqrt_scaled",' \n ',
      x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
    )

    #Specify survey design to run clustered analyses (cluster = family ID)
    survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

    #Fit the model specified above
    CLPM_genr_sens1_fit <- lavaan(CLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
    #Perform complex sampling analysis
    CLPM_genr_sens1_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_sens1_fit,
                                         survey.design = survey_design_genr)
    #Store coefficients of interest
```

```
    #Which brain region
    results_CLPM_genr_sens1_m3_t1w[(rowcount_clpm+2),1] <- x
    #Which CBCL syndrome scale
    results_CLPM_genr_sens1_m3_t1w[(rowcount_clpm+2),2] <- a
    #Which time point
    results_CLPM_genr_sens1_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),3] <-
      c("T1-T2", "T2-T3")
    #Cross-sectional associations T1
    results_CLPM_genr_sens1_m3_t1w[c((rowcount_clpm+2)),4:6] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(31),c(5,6,8)]
    #Autoregressive parameters
    #CBCL
    results_CLPM_genr_sens1_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),7:9] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
    #MRI
    results_CLPM_genr_sens1_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),10:12] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
    #Cross-lagged parameters
    #CBCL -> MRI
    results_CLPM_genr_sens1_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),13:15] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
    #MRI -> CBCL
    results_CLPM_genr_sens1_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),16:18] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
    #Which measure (actually only needs to be specified once)
    results_CLPM_genr_sens1_m3_t1w[1,c(6,8,11,14,17)] <-
      c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
    results_CLPM_genr_sens1_m3_t1w[2,] <- c("Brain region", "Syndrome Scale", "Wave",
                                            "B", "S.E.", "p-value",
                                            "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                            "B", "S.E.", "p-value", "B", "S.E.", "p-value")
    #Store summary statistics
    fitmeaures_CLPM_genr_sens1_m3_t1w[rowcount_fit, 1:5] <-
      c(x, summary(CLPM_genr_sens1_CSA, fit.measures = T,
                   standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
    #Store raw Rsquared
    rsquared_CLPM_genr_sens1_m3_t1w[rowcount_fit, 1:7] <-
      c(x, lavInspect(CLPM_genr_sens1_CSA, "rsquare"))
    #Adapt rowcounts to make sure results are stored properly
    rowcount_clpm <- rowcount_clpm + 2
    rowcount_fit <- rowcount_fit + 1
  }
}

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#DTI
#Specify the model for all brain morphology measures
for(a in cbcl_scales_genr){
  for(x in sig_regions_genr_clpm_dti){
    CLPM_genr <- paste0(
```

```
    ' \n
# Estimate the lagged effects between the variables
  ', a, "_9m_sqrt_scaled",' +', x,"_f09_scaled", ' ~
  ', a, "_5_sqrt_scaled",' +', x,"_f05_scaled",
    ' \n ' , a, "_14_sqrt_scaled",' +', x,"_f13_scaled", ' ~
  ', a, "_9m_sqrt_scaled",' +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n', a, "_5_sqrt_scaled",' ~ sex +  maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
    x,"_f05_scaled", ' ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

  #Estimate time dependent predictors
  ', a, "_5_sqrt_scaled",' ~ agechild_GR1075 \n
  ', a, "_9m_sqrt_scaled",' ~ AgeChild_CBCL9m \n
  ', a, "_14_sqrt_scaled",' ~ AGECHILD_GR1093 \n ',
    x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

  # Estimate the covariance between variables at the first wave.
  ', a, "_5_sqrt_scaled",' ~~ ', x,"_f05_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n ', a, "_9m_sqrt_scaled",' ~~ ', x,"_f09_scaled",
    ' \n ', a, "_14_sqrt_scaled",' ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n ', a, "_5_sqrt_scaled",' ~~ ', a, "_5_sqrt_scaled",' \n ', # Variances
    x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
    ' \n ', a, "_9m_sqrt_scaled",' ~~ ', a, "_9m_sqrt_scaled",' \n ', # Residual variances
    x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
    ' \n ', a, "_14_sqrt_scaled",' ~~ ', a, "_14_sqrt_scaled",' \n ',
    x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

  #Fit the model specified above
  CLPM_genr_sens1_fit <- lavaan(CLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_genr_sens1_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_sens1_fit,
                                       survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_genr_sens1_m2_dti[(rowcount_clpm+2),1] <- x
  #Which CBCL syndrome scale
  results_CLPM_genr_sens1_m2_dti[(rowcount_clpm+2),2] <- a
  #Which time point
  results_CLPM_genr_sens1_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),3] <- c("T1-T2", "T2-T3")
  #Cross-sectional associations T1
```

```
    results_CLPM_genr_sens1_m2_dti[c((rowcount_clpm+2)),4:6] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(28),c(5,6,8)]
    #Autoregressive parameters
    #CBCL
    results_CLPM_genr_sens1_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),7:9] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
    #MRI
    results_CLPM_genr_sens1_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),10:12] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
    #Cross-lagged parameters
    #CBCL -> MRI
    results_CLPM_genr_sens1_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),13:15] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
    #MRI -> CBCL
    results_CLPM_genr_sens1_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),16:18] <-
      summary(CLPM_genr_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
    #Which measure (actually only needs to be specified once)
    results_CLPM_genr_sens1_m2_dti[1,c(6,8,11,14,17)] <-
      c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
    results_CLPM_genr_sens1_m2_dti[2,] <- c("Brain region", "Syndrome Scale", "Wave",
                                            "B", "S.E.", "p-value",
                                            "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                            "B", "S.E.", "p-value", "B", "S.E.", "p-value")
    #Store summary statistics
    fitmeaures_CLPM_genr_sens1_m2_dti[rowcount_fit, 1:5] <-
      c(x, summary(CLPM_genr_sens1_CSA, fit.measures = T,
                   standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
    #Store raw Rsquared
    rsquared_CLPM_genr_sens1_m2_dti[rowcount_fit, 1:7] <-
      c(x, lavInspect(CLPM_genr_sens1_CSA, "rsquare"))
    #Adapt rowcounts to make sure results are stored properly
    rowcount_clpm <- rowcount_clpm + 2
    rowcount_fit <- rowcount_fit + 1
  }
}

#RICLPM
#Specify rowcounts
rowcount_riclpm <- 1
rowcount_fit <- 1

for(a in cbcl_scales_genr){
  for(x in sig_regions_genr_riclpm_t1w){
    RICLPM_genr <- paste0('
  # Create between components (random intercepts)
  RIx =~ 1*', a, '_5_sqrt_scaled + 1*', a, '_9m_sqrt_scaled + 1*', a, '_14_sqrt_scaled
  RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled", ' \n + 1*', x,"_f13_scaled",

# Create within-person centered variables
' \n  wx1 =~ 1*', a, '_5_sqrt_scaled
  wx2 =~ 1*', a, '_9m_sqrt_scaled
  wx3 =~ 1*', a, '_14_sqrt_scaled
  wy1 =~ 1* \n', x,"_f05_scaled",
```

```
' \n wy2 =~ 1* \n', x,"_f09_scaled",
' \n wy3 =~ 1* \n', x,"_f13_scaled",

# Estimate the lagged effects between the variables
' \n wx2 + wy2 ~ wx1 + wy1
  wx3 + wy3 ~ wx2 + wy2

  #Estimate time independent predictors
  RIx ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes
  RIy ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes

  #Estimate time dependent predictors
  wx1 ~ agechild_GR1075
  wx2 ~ AgeChild_CBCL9m
  wx3 ~ AGECHILD_GR1093
  wy1 ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled
  wy2 ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled
  wy3 ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled

  # Estimate the covariance between variables at the first wave.
  wx1 ~~ wy1 # Covariance

  # Estimate the covariances between the residuals of variables
  wx2 ~~ wy2
  wx3 ~~ wy3

  # Estimate the variance and covariance of the random intercepts.
  RIx ~~ RIx
  RIy ~~ RIy
  RIx ~~ RIy

  # Estimate the (residual) variance of the within-person centered variables.
  wx1 ~~ wx1 # Variances
  wy1 ~~ wy1
  wx2 ~~ wx2 # Residual variances
  wy2 ~~ wy2
  wx3 ~~ wx3
  wy3 ~~ wy3
  ')

    #Specify survey design to run clustered analyses (cluster = family ID)
    survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

    #Fit the model specified above
    RICLPM_genr_fit <- lavaan(RICLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
    #Perform complex sampling analysis
    RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                     survey.design = survey_design_genr)
    #Store coefficients of interest
    #Which brain region
    results_RICLPM_genr_sens1_m3_t1w[(rowcount_riclpm+2),1] <- x
```

```
    #Which CBCL syndrome scale
    results_RICLPM_genr_sens1_m3_t1w[(rowcount_riclpm+2),2] <- a
    #Which time point
    results_RICLPM_genr_sens1_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),3] <-
        c("T1-T2", "T2-T3")
    #Cross-sectional associations T1
    results_RICLPM_genr_sens1_m3_t1w[c((rowcount_riclpm+2)),4:6] <-
      summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(43),c(5,6,8)]
    #Autoregressive parameters
    #CBCL
    results_RICLPM_genr_sens1_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),7:9] <-
      summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(13,17),c(5,6,8)]
    #MRI
    results_RICLPM_genr_sens1_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),10:12] <-
      summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(16,20),c(5,6,8)]
    #Cross-lagged parameters
    #CBCL -> MRI
    results_RICLPM_genr_sens1_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),13:15] <-
      summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(15,19),c(5,6,8)]
    #MRI -> CBCL
    results_RICLPM_genr_sens1_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),16:18] <-
      summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14,18),c(5,6,8)]
    #Which measure (actually only needs to be specified once)
    results_RICLPM_genr_sens1_m3_t1w[1,c(5,8,11,14,17)] <-
      c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
    results_RICLPM_genr_sens1_m3_t1w[2,] <-
        c("Brain region", "Syndrome Scale", "Wave", "B", "S.E.", "p-value",
                                        "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                        "B", "S.E.", "p-value", "B", "S.E.", "p-value")
    #Store summary statistics
    fitmeaures_RICLPM_genr_sens1_m3_t1w[rowcount_fit, 1:4] <-
      summary(RICLPM_genr_CSA, fit.measures = T,
      standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")]
    #Store raw Rsquared
    rsquared_RICLPM_genr_sens1_m3_t1w[rowcount_fit, 1:9] <-
      c(x, lavInspect(RICLPM_genr_CSA, "rsquare")[1:8])
    #Adapt rowcounts to make sure results are stored properly
    rowcount_riclpm <- rowcount_riclpm + 2
    rowcount_fit <- rowcount_fit + 1
  }
}

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_genr_sens1_m3_t1w, "results_CLPM_genr_sens1_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_sens1_m3_t1w, "fitmeaures_CLPM_genr_sens1_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_sens1_m3_t1w, "rsquared_CLPM_genr_sens1_m3_t1w.csv",
          row.names = F, quote = F)

write.csv(results_CLPM_genr_sens1_m2_dti, "results_CLPM_genr_sens1_m2_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_sens1_m2_dti, "fitmeaures_CLPM_genr_sens1_m2_dti.csv",
```

```
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_sens1_m2_dti, "rsquared_CLPM_genr_sens1_m2_dti.csv",
          row.names = F, quote = F)

write.csv(results_RICLPM_genr_sens1_m3_t1w, "results_RICLPM_genr_sens1_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_RICLPM_genr_sens1_m3_t1w, "fitmeaures_RICLPM_genr_sens1_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_RICLPM_genr_sens1_m3_t1w, "rsquared_RICLPM_genr_sens1_m3_t1w.csv",
          row.names = F, quote = F)
```

4.1.2 ABCD Study

```
#Define CBCL syndrome scales
#Suffixes will be added within CLPM
cbcl_scales_abcd <-
 c("cbcl_scr_syn_anxdep_r", "cbcl_scr_syn_attention_r", "cbcl_scr_syn_aggressive_r")

#Create empty dataframes to store results
results_CLPM_abcd_sens1_m3_t1w <- data.frame()
fitmeaures_CLPM_abcd_sens1_m3_t1w <- data.frame()
rsquared_CLPM_abcd_sens1_m3_t1w <- data.frame()

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
for(a in cbcl_scales_abcd){
  for(x in sig_regions_abcd_clpm_t1w){
    CLPM_abcd <- paste0(
      ' \n
  # Estimate the lagged effects between the variables
    ', a, "_2_year_follow_up_y_arm_1_sqrt_scaled",' +', x,"_2_year_follow_up_y_arm_1_scaled", ' ~
    ', a, "_baseline_year_1_arm_1_sqrt_scaled",' +', x,"_baseline_year_1_arm_1_scaled",

      #Estimate time independent predictors
      ' \n \n ', a, "_baseline_year_1_arm_1_sqrt_scaled",' ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian \n ',
      x,"_baseline_year_1_arm_1_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian +
    site1 + site2 + site3 + site4 + site5 + site6 + site7 + site 8 + site9 + site10 +
    site11 + site12 + site13 + site14 + site15 + site16 + site17 + site18 + site19 +
    site20
    \n \n

    #Estimate time dependent predictors
    ', a, "_baseline_year_1_arm_1_sqrt_scaled",' ~ interview_age_baseline_year_1_arm_1 \n
    ', a, "_2_year_follow_up_y_arm_1_sqrt_scaled",' ~ interview_age_2_year_follow_up_y_arm_1 \n ',
      x,"_baseline_year_1_arm_1_scaled", ' ~ interview_age_baseline_year_1_arm_1 + handedness +
    smri_vol_scs_intracranialv_baseline_year_1_arm_1_scaled \n ',
```

```
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~ interview_age_2_year_follow_up_y_arm_1 + handedness +
smri_vol_scs_intracranialv_2_year_follow_up_y_arm_1_scaled \n

# Estimate the covariance between variables at the first wave.
', a, "_baseline_year_1_arm_1_sqrt_scaled",' ~~ ', x,"_baseline_year_1_arm_1_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n ', a, "_2_year_follow_up_y_arm_1_sqrt_scaled",' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n ', a, "_baseline_year_1_arm_1_sqrt_scaled",' ~~ ', a, "_baseline_year_1_arm_1_sqrt_scaled",'
  x,"_baseline_year_1_arm_1_scaled", ' ~~ ', x,"_baseline_year_1_arm_1_scaled",
  ' \n ', a, "_2_year_follow_up_y_arm_1_sqrt_scaled",' ~~ ',
  a, "_2_year_follow_up_y_arm_1_sqrt_scaled",' \n ', # Residual variances
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_abcd = svydesign(id = ~rel_family_id_baseline_year_1_arm_1, data = data_abcd_t1w)

#Fit the model specified above
CLPM_abcd_sens1_fit <- lavaan(CLPM_abcd, data = implist_t1w_abcd[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_abcd_sens1_CSA <- lavaan.survey(lavaan.fit = CLPM_abcd_sens1_fit,
                                     survey.design = survey_design_abcd)
#Store coefficients of interest
#Which brain region
results_CLPM_abcd_sens1_m3_t1w[(rowcount_clpm+2),1] <- x
#Which CBCL syndrome scale
results_CLPM_abcd_sens1_m3_t1w[(rowcount_clpm+2),2] <- a
#Cross-sectional associations T1
results_CLPM_abcd_sens1_m3_t1w[c((rowcount_clpm+2)),3:5] <-
  summary(CLPM_abcd_sens1_CSA, fit.measures = T, standardized = T)[[2]][c(48),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_abcd_sens1_m3_t1w[c((rowcount_clpm+2)),6:8] <-
  summary(CLPM_abcd_sens1_CSA, fit.measures = T, standardized = T)[[2]][1,c(5,6,8)]
#MRI
results_CLPM_abcd_sens1_m3_t1w[c((rowcount_clpm+2)),9:11] <-
  summary(CLPM_abcd_sens1_CSA, fit.measures = T, standardized = T)[[2]][4,c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_CLPM_abcd_sens1_m3_t1w[c((rowcount_clpm+2)),12:14] <-
  summary(CLPM_abcd_sens1_CSA, fit.measures = T, standardized = T)[[2]][3,c(5,6,8)]
#MRI -> CBCL
results_CLPM_abcd_sens1_m3_t1w[c((rowcount_clpm+2)),15:17] <-
  summary(CLPM_abcd_sens1_CSA, fit.measures = T, standardized = T)[[2]][2,c(5,6,8)]
#Which measure (actually only needs to be specified once)
results_CLPM_abcd_sens1_m3_t1w[1,c(4,7,10,13,16)] <-
  c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
results_CLPM_abcd_sens1_m3_t1w[2,] <- c("Brain region", "Syndrome Scale", "B", "S.E.", "p-value",
                                        "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                        "B", "S.E.", "p-value", "B", "S.E.", "p-value")
```

```
    #Store summary statistics
    fitmeaures_CLPM_abcd_sens1_m3_t1w[rowcount_fit, 1:5] <-
      c(x, summary(CLPM_abcd_sens1_CSA, fit.measures = T,
                   standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
    #Store raw Rsquared
    rsquared_CLPM_abcd_sens1_m3_t1w[rowcount_fit, 1:5] <-
      c(x, lavInspect(CLPM_abcd_sens1_CSA, "rsquare"))
    #Adapt rowcounts to make sure results are stored properly
    rowcount_clpm <- rowcount_clpm + 1
    rowcount_fit <- rowcount_fit + 1
  }
}

#Calculate relative Rsquared
  #First calculate mean Rsquared
  mean_rsquared_abcd_sens1_m3_t1w <-
    mean(as.numeric(c(rsquared_CLPM_abcd_sens1_m3_t1w[,2],
                      rsquared_CLPM_abcd_sens1_m3_t1w[,3],
                      rsquared_CLPM_abcd_sens1_m3_t1w[,4],
                      rsquared_CLPM_abcd_sens1_m3_t1w[,5])))
  #Then subtract mean from all Rsquared values to
  #obtain the relative Rsquared values
  rsquared_CLPM_abcd_sens1_m3_t1w[,"relative_DP_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_sens1_m3_t1w[,2])-
                 mean_rsquared_abcd_sens1_m3_t1w)
  rsquared_CLPM_abcd_sens1_m3_t1w[,"relative_MRI_T1"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_sens1_m3_t1w[,3])-
                 mean_rsquared_abcd_sens1_m3_t1w)
  rsquared_CLPM_abcd_sens1_m3_t1w[,"relative_DP_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_sens1_m3_t1w[,4])-
                 mean_rsquared_abcd_sens1_m3_t1w)
  rsquared_CLPM_abcd_sens1_m3_t1w[,"relative_MRI_T2"] <-
    as.numeric(as.numeric(rsquared_CLPM_abcd_sens1_m3_t1w[,5])-
                 mean_rsquared_abcd_sens1_m3_t1w)

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_abcd_sens1_m3_t1w, "results_CLPM_abcd_sens1_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_abcd_sens1_m3_t1w, "fitmeaures_CLPM_abcd_sens1_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_abcd_sens1_m3_t1w, "rsquared_CLPM_abcd_sens1_m3_t1w.csv",
          row.names = F, quote = F)
```

4.2 CBCL total problems score

In the second sensitivity analysis, we re-analyze the cross-lagged panel models that showed a significant effect in the cross-lagged paths, with the total problems scale of the CBCL, to assess to what extent the findings are specific to the CBCL-DP or are more broadly associated to broader psychopathology symptoms.

4.2.1 Generation R

```
#Create empty dataframes to store results
results_CLPM_genr_sens2_m3_t1w <- data.frame()
fitmeaures_CLPM_genr_sens2_m3_t1w <- data.frame()
rsquared_CLPM_genr_sens2_m3_t1w <- data.frame()

results_CLPM_genr_sens2_m2_dti <- data.frame()
fitmeaures_CLPM_genr_sens2_m2_dti <- data.frame()
rsquared_CLPM_genr_sens2_m2_dti <- data.frame()

results_RICLPM_genr_sens2_m3_t1w <- data.frame()
fitmeaures_RICLPM_genr_sens2_m3_t1w <- data.frame()
rsquared_RICLPM_genr_sens2_m3_t1w <- data.frame()

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#T1w
#Model 3
#Specify the model for all brain morphology measures
for(x in sig_regions_genr_clpm_t1w){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    cbcl_sum_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    cbcl_sum_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n cbcl_sum_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    cbcl_sum_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n cbcl_sum_5_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
    x,"_f05_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

    #Estimate time dependent predictors
    cbcl_sum_5_sqrt_scaled ~ agechild_GR1075 \n
    cbcl_sum_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    cbcl_sum_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled \n

    # Estimate the covariance between variables at the first wave.
    cbcl_sum_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n cbcl_sum_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
    ' \n cbcl_sum_14_sqrt_scaled ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n cbcl_sum_5_sqrt_scaled ~~ cbcl_sum_5_sqrt_scaled \n ', # Variances
    x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
```

```
    ' \n cbcl_sum_9m_sqrt_scaled ~~ cbcl_sum_9m_sqrt_scaled \n ', # Residual variances
    x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
    ' \n cbcl_sum_14_sqrt_scaled ~~ cbcl_sum_14_sqrt_scaled \n ',
    x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

  #Fit the model specified above
  CLPM_genr_fit <- lavaan(CLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                                 survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_genr_sens2_m3_t1w[(rowcount_clpm+2),1] <- x
  #Which time point
  results_CLPM_genr_sens2_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <- c("T1-T2", "T2-T3")
  #Cross-sectional associations T1
  results_CLPM_genr_sens2_m3_t1w[c((rowcount_clpm+2)),3:5] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(31),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_genr_sens2_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
  #MRI
  results_CLPM_genr_sens2_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_genr_sens2_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_sens2_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_sens2_m3_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_sens2_m3_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_sens2_m3_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_sens2_m3_t1w[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}
```

```
#DTI
#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
for(x in sig_regions_genr_clpm_dti){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    cbcl_sum_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    cbcl_sum_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n cbcl_sum_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    cbcl_sum_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n cbcl_sum_5_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
    x,"_f05_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

    #Estimate time dependent predictors
    cbcl_sum_5_sqrt_scaled ~ agechild_GR1075 \n
    cbcl_sum_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    cbcl_sum_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

    # Estimate the covariance between variables at the first wave.
    cbcl_sum_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n cbcl_sum_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
    ' \n cbcl_sum_14_sqrt_scaled ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n cbcl_sum_5_sqrt_scaled ~~ cbcl_sum_5_sqrt_scaled \n ', # Variances
    x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
    ' \n cbcl_sum_9m_sqrt_scaled ~~ cbcl_sum_9m_sqrt_scaled \n ', # Residual variances
    x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
    ' \n cbcl_sum_14_sqrt_scaled ~~ cbcl_sum_14_sqrt_scaled \n ',
    x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

  #Fit the model specified above
  CLPM_genr_fit <- lavaan(CLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                                 survey.design = survey_design_genr)
```

```
  #Store coefficients of interest
  #Which brain region
  results_CLPM_genr_sens2_m2_dti[(rowcount_clpm+2),1] <- x
  #Which time point
  results_CLPM_genr_sens2_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <-
    c("T1-T2", "T2-T3")
  #Cross-sectional associations T1
  results_CLPM_genr_sens2_m2_dti[c((rowcount_clpm+2)),3:5] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(28),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_genr_sens2_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
  #MRI
  results_CLPM_genr_sens2_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_genr_sens2_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_sens2_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_sens2_m2_dti[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_sens2_m2_dti[2,] <-
    c("Brain region", "Wave", "B", "S.E.", "p-value",
      "B", "S.E.", "p-value", "B", "S.E.", "p-value",
      "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_sens2_m2_dti[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_sens2_m2_dti[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#RICLPM
#Reset rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
for(x in sig_regions_genr_riclpm_t1w){
  RICLPM_genr <- paste0('
  # Create between components (random intercepts)
  RIx =~ 1*cbcl_sum_5_sqrt_scaled + 1*cbcl_sum_9m_sqrt_scaled + 1*cbcl_sum_14_sqrt_scaled
  RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled", ' \n + 1*', x,"_f13_scaled",
```

```
# Create within-person centered variables
' \n  wx1 =~ 1*cbcl_sum_5_sqrt_scaled
wx2 =~ 1*cbcl_sum_9m_sqrt_scaled
wx3 =~ 1*cbcl_sum_14_sqrt_scaled
wy1 =~ 1* \n', x,"_f05_scaled",
' \n wy2 =~ 1* \n', x,"_f09_scaled",
' \n wy3 =~ 1* \n', x,"_f13_scaled",

# Estimate the lagged effects between the variables
' \n wx2 + wy2 ~ wx1 + wy1
wx3 + wy3 ~ wx2 + wy2

#Estimate time independent predictors
RIx ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes
RIy ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes

#Estimate time dependent predictors
wx1 ~ agechild_GR1075
wx2 ~ AgeChild_CBCL9m
wx3 ~ AGECHILD_GR1093
wy1 ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled
wy2 ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled
wy3 ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled

# Estimate the covariance between variables at the first wave.
wx1 ~~ wy1 # Covariance

# Estimate the covariances between the residuals of variables
wx2 ~~ wy2
wx3 ~~ wy3

# Estimate the variance and covariance of the random intercepts.
RIx ~~ RIx
RIy ~~ RIy
RIx ~~ RIy

# Estimate the (residual) variance of the within-person centered variables.
wx1 ~~ wx1 # Variances
wy1 ~~ wy1
wx2 ~~ wx2 # Residual variances
wy2 ~~ wy2
wx3 ~~ wx3
wy3 ~~ wy3
')

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
RICLPM_genr_fit <- lavaan(RICLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
```

```
  RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                   survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_RICLPM_genr_sens2_m3_t1w[(rowcount_riclpm+2),1] <- x
  #Which time point
  results_RICLPM_genr_sens2_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),2] <-
    c("T1-T2", "T2-T3")
  #Cross-sectional associations T1
  results_RICLPM_genr_sens2_m3_t1w[c((rowcount_riclpm+2)),3:5] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(43),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_RICLPM_genr_sens2_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),6:8] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(13,17),c(5,6,8)]
  #MRI
  results_RICLPM_genr_sens2_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),9:11] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(16,20),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_RICLPM_genr_sens2_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),12:14] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(15,19),c(5,6,8)]
  #MRI -> CBCL
  results_RICLPM_genr_sens2_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),15:17] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14,18),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_RICLPM_genr_sens2_m3_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_RICLPM_genr_sens2_m3_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_RICLPM_genr_sens2_m3_t1w[rowcount_fit, 1:4] <-
    summary(RICLPM_genr_CSA, fit.measures = T,
            standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")]
  #Store raw Rsquared
  rsquared_RICLPM_genr_sens2_m3_t1w[rowcount_fit, 1:9] <-
    c(x, lavInspect(RICLPM_genr_CSA, "rsquare")[1:8])
  #Adapt rowcounts to make sure results are stored properly
  rowcount_riclpm <- rowcount_riclpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_genr_sens2_m3_t1w, "results_CLPM_genr_sens2_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_sens2_m3_t1w, "fitmeaures_CLPM_genr_sens2_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_sens2_m3_t1w, "rsquared_CLPM_genr_sens2_m3_t1w.csv",
          row.names = F, quote = F)

write.csv(results_CLPM_genr_sens2_m2_dti, "results_CLPM_genr_sens2_m2_dti.csv",
```

```
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_sens2_m2_dti, "fitmeaures_CLPM_genr_sens2_m2_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_sens2_m2_dti, "rsquared_CLPM_genr_sens2_m2_dti.csv",
          row.names = F, quote = F)

write.csv(results_RICLPM_genr_sens2_m3_t1w, "results_RICLPM_genr_sens2_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_RICLPM_genr_sens2_m3_t1w, "fitmeaures_RICLPM_genr_sens2_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_RICLPM_genr_sens2_m3_t1w, "rsquared_RICLPM_genr_sens2_m3_t1w.csv",
          row.names = F, quote = F)
```

4.2.2 ABCD Study

```
#Create empty dataframes to store results
results_CLPM_abcd_sens2_m3_t1w <- data.frame()
fitmeaures_CLPM_abcd_sens2_m3_t1w <- data.frame()
rsquared_CLPM_abcd_sens2_m3_t1w <- data.frame()

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#Model 3
#Specify the model for all brain morphology measures
#Only correct volumes for ICV, not DTI measures
#T1-weighted
for(x in sig_regions_abcd_clpm_t1w){
  CLPM_abcd <- paste0(
    '
    # Estimate the lagged effects between the variables
    cbcl_scr_syn_totprob_r_2_year_follow_up_y_arm_1_sqrt_scaled +',
    x,"_2_year_follow_up_y_arm_1_scaled", ' ~
    cbcl_scr_syn_totprob_r_baseline_year_1_arm_1_sqrt_scaled +',
    x,"_baseline_year_1_arm_1_scaled",

    #Estimate time independent predictors
    ' \n \n cbcl_scr_syn_totprob_r_baseline_year_1_arm_1_sqrt_scaled ~ sex +
    maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian +
    site1 + site2 + site3 + site4 + site5 + site6 + site7 + site 8 + site9 + site10 +
    site11 + site12 + site13 + site14 + site15 + site16 + site17 + site18 + site19 +
    site20
    \n \n

    #Estimate time dependent predictors
    cbcl_scr_syn_totprob_r_baseline_year_1_arm_1_sqrt_scaled ~
    interview_age_baseline_year_1_arm_1 \n
```

```
  cbcl_scr_syn_totprob_r_2_year_follow_up_y_arm_1_sqrt_scaled ~
  interview_age_2_year_follow_up_y_arm_1 \n ',
  x,"_baseline_year_1_arm_1_scaled", ' ~ interview_age_baseline_year_1_arm_1 +
  handedness +
  smri_vol_scs_intracranialv_baseline_year_1_arm_1_scaled \n ',
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~ interview_age_2_year_follow_up_y_arm_1 +
  handedness +
  smri_vol_scs_intracranialv_2_year_follow_up_y_arm_1_scaled \n

  # Estimate the covariance between variables at the first wave.
  cbcl_scr_syn_totprob_r_baseline_year_1_arm_1_sqrt_scaled ~~ '
  , x,"_baseline_year_1_arm_1_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n cbcl_scr_syn_totprob_r_2_year_follow_up_y_arm_1_sqrt_scaled ~~ '
  , x,"_2_year_follow_up_y_arm_1_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n cbcl_scr_syn_totprob_r_baseline_year_1_arm_1_sqrt_scaled ~~
  cbcl_scr_syn_totprob_r_baseline_year_1_arm_1_sqrt_scaled \n ', # Variances
  x,"_baseline_year_1_arm_1_scaled", ' ~~ ', x,"_baseline_year_1_arm_1_scaled",
  ' \n cbcl_scr_syn_totprob_r_2_year_follow_up_y_arm_1_sqrt_scaled ~~
  cbcl_scr_syn_totprob_r_2_year_follow_up_y_arm_1_sqrt_scaled \n ', # Residual variances
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_abcd <-
  svydesign(id = ~rel_family_id_baseline_year_1_arm_1, data = data_abcd_t1w)

#Fit the model specified above
CLPM_abcd_fit <- lavaan(CLPM_abcd, data = implist_t1w_abcd[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_abcd_CSA <- lavaan.survey(lavaan.fit = CLPM_abcd_fit,
                               survey.design = survey_design_abcd)
#Store coefficients of interest
#Which brain region
results_CLPM_abcd_sens2_m3_t1w[(rowcount_clpm+2),1] <- x
#Cross-sectional associations T1
results_CLPM_abcd_sens2_m3_t1w[c((rowcount_clpm+2)),2:4] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][c(48),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_abcd_sens2_m3_t1w[c((rowcount_clpm+2)),5:7] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][1,c(5,6,8)]
#MRI
results_CLPM_abcd_sens2_m3_t1w[c((rowcount_clpm+2)),8:10] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][4,c(5,6,8)]
#Cross-lagged parameters
#CBCL -> MRI
results_CLPM_abcd_sens2_m3_t1w[c((rowcount_clpm+2)),11:13] <-
  summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][3,c(5,6,8)]
#MRI -> CBCL
```

```
  results_CLPM_abcd_sens2_m3_t1w[c((rowcount_clpm+2)),14:16] <-
    summary(CLPM_abcd_CSA, fit.measures = T, standardized = T)[[2]][2,c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_abcd_sens2_m3_t1w[1,c(3,6,9,12,15)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_abcd_sens2_m3_t1w[2,] <- c("Brain region", "B", "S.E.", "p-value",
                                          "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                          "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_abcd_sens2_m3_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_abcd_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_abcd_sens2_m3_t1w[rowcount_fit, 1:5] <-
    c(x, lavInspect(CLPM_abcd_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 1
  rowcount_fit <- rowcount_fit + 1
}

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_abcd_sens2_m3_t1w, "results_CLPM_abcd_sens2_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_abcd_sens2_m3_t1w, "fitmeaures_CLPM_abcd_sens2_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_abcd_sens2_m3_t1w, "rsquared_CLPM_abcd_sens2_m3_t1w.csv",
          row.names = F, quote = F)
```

4.3 Cognitive performance as covariate

As a third sensitivity analysis, we re-analyze the cross-lagged panel models that showed a significant effect in the cross-lagged paths, including cognitive performance as an additional covariate in the models.

4.3.1 Generation R

```
#Create empty dataframes to store results
results_CLPM_genr_sens3_m3_t1w <- data.frame()
fitmeaures_CLPM_genr_sens3_m3_t1w <- data.frame()
rsquared_CLPM_genr_sens3_m3_t1w <- data.frame()

results_CLPM_genr_sens3_m2_dti <- data.frame()
fitmeaures_CLPM_genr_sens3_m2_dti <- data.frame()
rsquared_CLPM_genr_sens3_m2_dti <- data.frame()

results_RICLPM_genr_sens3_m3_t1w <- data.frame()
fitmeaures_RICLPM_genr_sens3_m3_t1w <- data.frame()
rsquared_RICLPM_genr_sens3_m3_t1w <- data.frame()

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1
```

```
#T1w
#Model 3
#Specify the model for all brain morphology measures
for(x in sig_regions_genr_clpm_t1w){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_5_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes + WISC13_FSIQ \n ',
    x,"_f05_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes + WISC13_FSIQ \n

    #Estimate time dependent predictors
    sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
    sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
    x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
    ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
    x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
    x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

  #Fit the model specified above
  CLPM_genr_sens3_fit <- lavaan(CLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_genr_sens3_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_sens3_fit,
                               survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_genr_sens3_m3_t1w[(rowcount_clpm+2),1] <- x
```

```
  #Which time point
  results_CLPM_genr_sens3_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <-
    c("T1-T2", "T2-T3")
  #Cross-sectional associations T1
  results_CLPM_genr_sens3_m3_t1w[c((rowcount_clpm+2)),3:5] <-
    summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(33),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_genr_sens3_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
    summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
  #MRI
  results_CLPM_genr_sens3_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
    summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_genr_sens3_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_sens3_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_sens3_m3_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_sens3_m3_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_sens3_m3_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_sens3_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_sens3_m3_t1w[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_sens3_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#DTI
#Model 2
#Specify the model for all brain morphology measures
for(x in sig_regions_genr_clpm_dti){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",
```

```
  #Estimate time independent predictors
  ' \n \n sum_dp_5_sqrt_scaled ~ sex +  maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes + WISC13_FSIQ \n ',
  x,"_f05_scaled", ' ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes + WISC13_FSIQ \n

  #Estimate time dependent predictors
  sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
  sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
  sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
  x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 \n ',
  x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
  x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
  x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
  ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
  x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
  ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
  x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

#Fit the model specified above
CLPM_genr_sens3_fit <- lavaan(CLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_genr_sens3_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_sens3_fit,
                             survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_CLPM_genr_sens3_m2_dti[(rowcount_clpm+2),1] <- x
#Which time point
results_CLPM_genr_sens3_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <-
  c("T1-T2", "T2-T3")
#Cross-sectional associations T1
results_CLPM_genr_sens3_m2_dti[c((rowcount_clpm+2)),3:5] <-
  summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(30),c(5,6,8)]
#Autoregressive parameters
#CBCL
results_CLPM_genr_sens3_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
  summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
#MRI
results_CLPM_genr_sens3_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
```

```
    summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_genr_sens3_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_sens3_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_sens3_m2_dti[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_sens3_m2_dti[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_sens3_m2_dti[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_sens3_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_sens3_m2_dti[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_sens3_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Reset rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#RICLPM
#Model 3
#T1w
#Reset rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
for(x in sig_regions_genr_riclpm_t1w){
  RICLPM_genr <- paste0('
  # Create between components (random intercepts)
  RIx =~ 1*sum_dp_5_sqrt_scaled + 1*sum_dp_9m_sqrt_scaled + 1*sum_dp_14_sqrt_scaled
  RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled", ' \n + 1*', x,"_f13_scaled",

  # Create within-person centered variables
  ' \n  wx1 =~ 1*sum_dp_5_sqrt_scaled
  wx2 =~ 1*sum_dp_9m_sqrt_scaled
  wx3 =~ 1*sum_dp_14_sqrt_scaled
  wy1 =~ 1* \n', x,"_f05_scaled",
  ' \n wy2 =~ 1* \n', x,"_f09_scaled",
  ' \n wy3 =~ 1* \n', x,"_f13_scaled",

  # Estimate the lagged effects between the variables
```

```
' \n wx2 + wy2 ~ wx1 + wy1
wx3 + wy3 ~ wx2 + wy2

#Estimate time independent predictors
RIx ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes + WISC13_FSIQ
RIy ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes + WISC13_FSIQ

#Estimate time dependent predictors
wx1 ~ agechild_GR1075
wx2 ~ AgeChild_CBCL9m
wx3 ~ AGECHILD_GR1093
wy1 ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled
wy2 ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled
wy3 ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled

# Estimate the covariance between variables at the first wave.
wx1 ~~ wy1 # Covariance

# Estimate the covariances between the residuals of variables
wx2 ~~ wy2
wx3 ~~ wy3

# Estimate the variance and covariance of the random intercepts.
RIx ~~ RIx
RIy ~~ RIy
RIx ~~ RIy

# Estimate the (residual) variance of the within-person centered variables.
wx1 ~~ wx1 # Variances
wy1 ~~ wy1
wx2 ~~ wx2 # Residual variances
wy2 ~~ wy2
wx3 ~~ wx3
wy3 ~~ wy3
')

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
RICLPM_genr_fit <- lavaan(RICLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                 survey.design = survey_design_genr)
#Store coefficients of interest
#Which brain region
results_RICLPM_genr_sens3_m3_t1w[(rowcount_riclpm+2),1] <- x
#Which time point
results_RICLPM_genr_sens3_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),2] <-
  c("T1-T2", "T2-T3")
#Cross-sectional associations T1
```

```
  results_RICLPM_genr_sens3_m3_t1w[c((rowcount_riclpm+2)),3:5] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(45),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_RICLPM_genr_sens3_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),6:8] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(13,17),c(5,6,8)]
  #MRI
  results_RICLPM_genr_sens3_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),9:11] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(16,20),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_RICLPM_genr_sens3_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),12:14] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(15,19),c(5,6,8)]
  #MRI -> CBCL
  results_RICLPM_genr_sens3_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),15:17] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14,18),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_RICLPM_genr_sens3_m3_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_RICLPM_genr_sens3_m3_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                                  "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                  "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_RICLPM_genr_sens3_m3_t1w[rowcount_fit, 1:4] <-
    summary(RICLPM_genr_CSA, fit.measures = T,
            standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")]
  #Store raw Rsquared
  rsquared_RICLPM_genr_sens3_m3_t1w[rowcount_fit, 1:9] <-
    c(x, lavInspect(RICLPM_genr_CSA, "rsquare")[1:8])
  #Adapt rowcounts to make sure results are stored properly
  rowcount_riclpm <- rowcount_riclpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_genr_sens3_m3_t1w, "results_CLPM_genr_sens3_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_sens3_m3_t1w, "fitmeaures_CLPM_genr_sens3_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_sens3_m3_t1w, "rsquared_CLPM_genr_sens3_m3_t1w.csv",
          row.names = F, quote = F)

write.csv(results_CLPM_genr_sens3_m2_dti, "results_CLPM_genr_sens3_m2_dti.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_sens3_m2_dti, "fitmeaures_CLPM_genr_sens3_m2_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_sens3_m2_dti, "rsquared_CLPM_genr_sens3_m2_dti.csv",
          row.names = F, quote = F)

write.csv(results_RICLPM_genr_sens3_m3_t1w, "results_RICLPM_genr_sens3_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_RICLPM_genr_sens3_m3_t1w, "fitmeaures_RICLPM_genr_sens3_m3_t1w.csv",
```

```
          row.names = F, quote = F)
write.csv(rsquared_RICLPM_genr_sens3_m3_t1w, "rsquared_RICLPM_genr_sens3_m3_t1w.csv",
          row.names = F, quote = F)
```

4.3.2 ABCD Study

```
#Create empty dataframes to store results
results_CLPM_abcd_sens3_m3_t1w <- data.frame()
fitmeaures_CLPM_abcd_sens3_m3_t1w <- data.frame()
rsquared_CLPM_abcd_sens3_m3_t1w <- data.frame()

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1


#Specify the model for all brain morphology measures
for(x in sig_regions_abcd_clpm_t1w){
  CLPM_abcd <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_t2_sqrt_scaled +', x,"_2_year_follow_up_y_arm_1_scaled", ' ~
    sum_dp_t1_sqrt_scaled +', x,"_baseline_year_1_arm_1_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_t1_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
    child_nationalorigin_hispanic + child_nationalorigin_asian +
    site1 + site2 + site3 + site4 + site5 + site6 + site7 + site 8 + site9 + site10 +
    site11 + site12 + site13 + site14 + site15 + site16 + site17 + site18 + site19 +
    site20
    \n \n

    #Estimate time dependent predictors
    sum_dp_t1_sqrt_scaled ~ interview_age_baseline_year_1_arm_1 +
    nihtbx_totalcomp_uncorrected_baseline_year_1_arm_1 \n
    sum_dp_t2_sqrt_scaled ~ interview_age_2_year_follow_up_y_arm_1 +
    nihtbx_totalcomp_uncorrected_2_year_follow_up_y_arm_1 \n ',
    x,"_baseline_year_1_arm_1_scaled", ' ~ interview_age_baseline_year_1_arm_1 + handedness +
    smri_vol_scs_intracranialv_baseline_year_1_arm_1_scaled +
    nihtbx_totalcomp_uncorrected_baseline_year_1_arm_1 \n ',
    x,"_2_year_follow_up_y_arm_1_scaled", ' ~ interview_age_2_year_follow_up_y_arm_1 + handedness +
    smri_vol_scs_intracranialv_2_year_follow_up_y_arm_1_scaled +
    nihtbx_totalcomp_uncorrected_2_year_follow_up_y_arm_1 \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_t1_sqrt_scaled ~~ ', x,"_baseline_year_1_arm_1_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_t2_sqrt_scaled ~~ ', x,"_2_year_follow_up_y_arm_1_scaled",
```

```
    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_t1_sqrt_scaled ~~ sum_dp_t1_sqrt_scaled \n ', # Variances
    x,"_baseline_year_1_arm_1_scaled", ' ~~ ', x,"_baseline_year_1_arm_1_scaled",
    ' \n sum_dp_t2_sqrt_scaled ~~ sum_dp_t2_sqrt_scaled \n ', # Residual variances
    x,"_2_year_follow_up_y_arm_1_scaled", ' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_abcd = svydesign(id = ~rel_family_id_baseline_year_1_arm_1, data = data_abcd_t1w)

  #Fit the model specified above
  CLPM_abcd_sens3_fit <- lavaan(CLPM_abcd, data = implist_t1w_abcd[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_abcd_sens3_CSA <- lavaan.survey(lavaan.fit = CLPM_abcd_sens3_fit,
                                       survey.design = survey_design_abcd)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_abcd_sens3_m3_t1w[(rowcount_clpm+2),1] <- x
  #Cross-sectional associations T1
  results_CLPM_abcd_sens3_m3_t1w[c((rowcount_clpm+2)),2:4] <-
    summary(CLPM_abcd_sens3_CSA, fit.measures = T, standardized = T)[[2]][c(52),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_abcd_sens3_m3_t1w[c((rowcount_clpm+2)),5:7] <-
    summary(CLPM_abcd_sens3_CSA, fit.measures = T, standardized = T)[[2]][1,c(5,6,8)]
  #MRI
  results_CLPM_abcd_sens3_m3_t1w[c((rowcount_clpm+2)),8:10] <-
    summary(CLPM_abcd_sens3_CSA, fit.measures = T, standardized = T)[[2]][4,c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_abcd_sens3_m3_t1w[c((rowcount_clpm+2)),11:13] <-
    summary(CLPM_abcd_sens3_CSA, fit.measures = T, standardized = T)[[2]][3,c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_abcd_sens3_m3_t1w[c((rowcount_clpm+2)),14:16] <-
    summary(CLPM_abcd_sens3_CSA, fit.measures = T, standardized = T)[[2]][2,c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_abcd_sens3_m3_t1w[1,c(3,6,9,12,15)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_abcd_sens3_m3_t1w[2,] <- c("Brain region", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_abcd_sens3_m3_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_abcd_sens3_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_abcd_sens3_m3_t1w[rowcount_fit, 1:5] <-
    c(x, lavInspect(CLPM_abcd_sens3_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 1
  rowcount_fit <- rowcount_fit + 1
}
```

```r
#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_abcd_sens3_m3_t1w, "results_CLPM_abcd_sens3_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_abcd_sens3_m3_t1w, "fitmeaures_CLPM_abcd_sens3_m3_t1w.csv",
          row.names = F, quote = F)
```

4.4 Excluding children with medication use

In a fourth sensitivity analysis, we re-analyze the cross-lagged panel models that showed a significant effect in the cross-lagged paths, excluding those participants that are using psychotropic medication.

4.4.1 Generation R

```r
#Exclude those with medication use
##Make list of imputed datasets to feed to lavaan
implist_genr <- lapply(seq(dsImp_genr$m), function(im) complete(dsImp_genr, im))

##Create dummy variables for categorical variables
for(x in seq(dsImp_genr$m)){
  #Sex
  implist_genr[[x]]$sex <-
    ifelse(implist_genr[[x]]$GENDER == "girl", 1, 0)
  #Maternal education
  implist_genr[[x]]$maternal_education_high <-
    ifelse(implist_genr[[x]]$maternal_education == "high", 1, 0)
  implist_genr[[x]]$maternal_education_middle <-
    ifelse(implist_genr[[x]]$maternal_education == "middle", 1, 0)
  #Child national origin
  implist_genr[[x]]$child_nationalorigin_dutch <-
    ifelse(implist_genr[[x]]$child_nationalorigin == "Dutch", 1, 0)
  implist_genr[[x]]$child_nationalorigin_wes <-
    ifelse(implist_genr[[x]]$child_nationalorigin == "other western", 1, 0)
}

#Create new lists to store subsetted dataframes
implist2_t1w_genr <- list()
implist2_dti_genr <- list()

for(x in seq(dsImp_genr$m)){
  implist2_t1w_genr[[x]] <- subset(implist_genr[[x]], meduse != 1 &
                                     implist_genr[[x]]$nscans > 1)
}

for(x in seq(dsImp_genr$m)){
  implist2_dti_genr[[x]] <- subset(implist_genr[[x]], meduse != 1 &
                                     implist_genr[[x]]$nscans_dti > 1)
}

#Create imputationlist object
data_genr_t1w <- imputationList(implist2_t1w_genr)
data_genr_dti <- imputationList(implist2_dti_genr)
```

```
#Create empty dataframes to store results
results_CLPM_genr_sens4_m3_t1w <- data.frame()
fitmeaures_CLPM_genr_sens4_m3_t1w <- data.frame()
rsquared_CLPM_genr_sens4_m3_t1w <- data.frame()

results_CLPM_genr_sens4_m2_dti <- data.frame()
fitmeaures_CLPM_genr_sens4_m2_dti <- data.frame()
rsquared_CLPM_genr_sens4_m2_dti <- data.frame()

results_RICLPM_genr_sens4_m3_t1w <- data.frame()
fitmeaures_RICLPM_genr_sens4_m3_t1w <- data.frame()
rsquared_RICLPM_genr_sens4_m3_t1w <- data.frame()

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#CLPM
#DTI
for(x in sig_regions_genr_clpm_dti){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_5_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
    x,"_f05_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

    #Estimate time dependent predictors
    sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
    sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
    x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
    ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
```

```
    x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
    x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_dti)

  #Fit the model specified above
  CLPM_genr_fit <- lavaan(CLPM_genr, data = implist_dti_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                                 survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_CLPM_genr_sens4_m2_dti[(rowcount_clpm+2),1] <- x
  #Which time point
  results_CLPM_genr_sens4_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <-
    c("T1-T2", "T2-T3")
  #Cross-sectional associations T1
  results_CLPM_genr_sens4_m2_dti[c((rowcount_clpm+2)),3:5] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(28),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_genr_sens4_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
  #MRI
  results_CLPM_genr_sens4_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_genr_sens4_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_sens4_m2_dti[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_sens4_m2_dti[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_sens4_m2_dti[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_sens4_m2_dti[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_sens4_m2_dti[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}
```

```
#Reset rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#T1w
#Specify the model for all brain morphology measures
for(x in sig_regions_genr_clpm_t1w){
  CLPM_genr <- paste0(
    '
  # Estimate the lagged effects between the variables
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled", ' ~
    sum_dp_5_sqrt_scaled +', x,"_f05_scaled",
    ' \n sum_dp_14_sqrt_scaled +', x,"_f13_scaled", ' ~
    sum_dp_9m_sqrt_scaled +', x,"_f09_scaled",

    #Estimate time independent predictors
    ' \n \n sum_dp_5_sqrt_scaled ~ sex +  maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n ',
    x,"_f05_scaled", ' ~ sex + maternal_education_middle +
    maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes \n

    #Estimate time dependent predictors
    sum_dp_5_sqrt_scaled ~ agechild_GR1075 \n
    sum_dp_9m_sqrt_scaled ~ AgeChild_CBCL9m \n
    sum_dp_14_sqrt_scaled ~ AGECHILD_GR1093 \n ',
    x,"_f05_scaled", ' ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled \n ',
    x,"_f09_scaled", ' ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled \n ',
    x,"_f13_scaled", ' ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled \n

    # Estimate the covariance between variables at the first wave.
    sum_dp_5_sqrt_scaled ~~ ', x,"_f05_scaled", # Covariance

    # Estimate the covariances between the residuals of variables
    ' \n sum_dp_9m_sqrt_scaled ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ ', x,"_f13_scaled",

    # Estimate the (residual) variance of variables of interest
    ' \n sum_dp_5_sqrt_scaled ~~ sum_dp_5_sqrt_scaled \n ', # Variances
    x,"_f05_scaled", ' ~~ ', x,"_f05_scaled",
    ' \n sum_dp_9m_sqrt_scaled ~~ sum_dp_9m_sqrt_scaled \n ', # Residual variances
    x,"_f09_scaled", ' ~~ ', x,"_f09_scaled",
    ' \n sum_dp_14_sqrt_scaled ~~ sum_dp_14_sqrt_scaled \n ',
    x,"_f13_scaled", ' ~~ ', x,"_f13_scaled"
  )

  #Specify survey design to run clustered analyses (cluster = family ID)
  survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

  #Fit the model specified above
  CLPM_genr_fit <- lavaan(CLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
  #Perform complex sampling analysis
  CLPM_genr_CSA <- lavaan.survey(lavaan.fit = CLPM_genr_fit,
                                 survey.design = survey_design_genr)
```

```
  #Store coefficients of interest
  #Which brain region
  results_CLPM_genr_sens4_m3_t1w[(rowcount_clpm+2),1] <- x
  #Which time point
  results_CLPM_genr_sens4_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),2] <-
    c("T1-T2", "T2-T3")
  #Cross-sectional associations T1
  results_CLPM_genr_sens4_m3_t1w[c((rowcount_clpm+2)),3:5] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(31),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_CLPM_genr_sens4_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),6:8] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(1,5),c(5,6,8)]
  #MRI
  results_CLPM_genr_sens4_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),9:11] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(4,8),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_genr_sens4_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),12:14] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(3,7),c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_genr_sens4_m3_t1w[c((rowcount_clpm+2):(rowcount_clpm+3)),15:17] <-
    summary(CLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(2,6),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_genr_sens4_m3_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_genr_sens4_m3_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_genr_sens4_m3_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_genr_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Store raw Rsquared
  rsquared_CLPM_genr_sens4_m3_t1w[rowcount_fit, 1:7] <-
    c(x, lavInspect(CLPM_genr_CSA, "rsquare"))
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#RICLPM
#Reset rowcount to keep track of where we are in the loop
rowcount_riclpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
for(x in sig_regions_genr_riclpm_t1w){
  RICLPM_genr <- paste0('
  # Create between components (random intercepts)
  RIx =~ 1*sum_dp_5_sqrt_scaled + 1*sum_dp_9m_sqrt_scaled + 1*sum_dp_14_sqrt_scaled
  RIy =~ 1*', x,"_f05_scaled", '\n + 1*', x,"_f09_scaled", ' \n + 1*', x,"_f13_scaled",
```

```
# Create within-person centered variables
' \n  wx1 =~ 1*sum_dp_5_sqrt_scaled
wx2 =~ 1*sum_dp_9m_sqrt_scaled
wx3 =~ 1*sum_dp_14_sqrt_scaled
wy1 =~ 1* \n', x,"_f05_scaled",
' \n wy2 =~ 1* \n', x,"_f09_scaled",
' \n wy3 =~ 1* \n', x,"_f13_scaled",

# Estimate the lagged effects between the variables
' \n wx2 + wy2 ~ wx1 + wy1
wx3 + wy3 ~ wx2 + wy2

#Estimate time independent predictors
RIx ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes
RIy ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_dutch + child_nationalorigin_wes

#Estimate time dependent predictors
wx1 ~ agechild_GR1075
wx2 ~ AgeChild_CBCL9m
wx3 ~ AGECHILD_GR1093
wy1 ~ age_child_mri_f05 + HC12_F5 + eTIV_f05_scaled
wy2 ~ age_child_mri_f09 + HC12_F9 + eTIV_f09_scaled
wy3 ~ age_child_mri_f13 + HC12_F13 + eTIV_f13_scaled

# Estimate the covariance between variables at the first wave.
wx1 ~~ wy1 # Covariance

# Estimate the covariances between the residuals of variables
wx2 ~~ wy2
wx3 ~~ wy3

# Estimate the variance and covariance of the random intercepts.
RIx ~~ RIx
RIy ~~ RIy
RIx ~~ RIy

# Estimate the (residual) variance of the within-person centered variables.
wx1 ~~ wx1 # Variances
wy1 ~~ wy1
wx2 ~~ wx2 # Residual variances
wy2 ~~ wy2
wx3 ~~ wx3
wy3 ~~ wy3
')

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_genr = svydesign(id = ~IDM, data = data_genr_t1w)

#Fit the model specified above
RICLPM_genr_fit <- lavaan(RICLPM_genr, data = implist_t1w_genr[[1]], estimator = 'MLM')
#Perform complex sampling analysis
```

```
  RICLPM_genr_CSA <- lavaan.survey(lavaan.fit = RICLPM_genr_fit,
                                   survey.design = survey_design_genr)
  #Store coefficients of interest
  #Which brain region
  results_RICLPM_genr_sens4_m3_t1w[(rowcount_riclpm+2),1] <- x
  #Which time point
  results_RICLPM_genr_sens4_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),2] <-
    c("T1-T2", "T2-T3")
  #Cross-sectional associations T1
  results_RICLPM_genr_sens4_m3_t1w[c((rowcount_riclpm+2)),3:5] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(43),c(5,6,8)]
  #Autoregressive parameters
  #CBCL
  results_RICLPM_genr_sens4_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),6:8] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(13,17),c(5,6,8)]
  #MRI
  results_RICLPM_genr_sens4_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),9:11] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(16,20),c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_RICLPM_genr_sens4_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),12:14] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(15,19),c(5,6,8)]
  #MRI -> CBCL
  results_RICLPM_genr_sens4_m3_t1w[c((rowcount_riclpm+2):(rowcount_riclpm+3)),15:17] <-
    summary(RICLPM_genr_CSA, fit.measures = T, standardized = T)[[2]][c(14,18),c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_RICLPM_genr_sens4_m3_t1w[1,c(4,7,10,13,16)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_RICLPM_genr_sens4_m3_t1w[2,] <- c("Brain region", "Wave", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                                    "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_RICLPM_genr_sens4_m3_t1w[rowcount_fit, 1:4] <-
    summary(RICLPM_genr_CSA, fit.measures = T,
            standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")]
  #Store raw Rsquared
  rsquared_RICLPM_genr_sens4_m3_t1w[rowcount_fit, 1:9] <-
    c(x, lavInspect(RICLPM_genr_CSA, "rsquare")[1:8])
  #Adapt rowcounts to make sure results are stored properly
  rowcount_riclpm <- rowcount_riclpm + 2
  rowcount_fit <- rowcount_fit + 1
}

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_genr_sens4_m3_t1w, "results_CLPM_genr_sens4_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_sens4_m3_t1w, "fitmeaures_CLPM_genr_sens4_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_sens4_m3_t1w, "rsquared_CLPM_genr_sens4_m3_t1w.csv",
          row.names = F, quote = F)

write.csv(results_CLPM_genr_sens4_m2_dti, "results_CLPM_genr_sens4_m2_dti.csv",
```

```
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_genr_sens4_m2_dti, "fitmeaures_CLPM_genr_sens4_m2_dti.csv",
          row.names = F, quote = F)
write.csv(rsquared_CLPM_genr_sens4_m2_dti, "rsquared_CLPM_genr_sens4_m2_dti.csv",
          row.names = F, quote = F)

write.csv(results_RICLPM_genr_sens4_m3_t1w, "results_RICLPM_genr_sens4_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_RICLPM_genr_sens4_m3_t1w, "fitmeaures_RICLPM_genr_sens4_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(rsquared_RICLPM_genr_sens4_m3_t1w, "rsquared_RICLPM_genr_sens4_m3_t1w.csv",
          row.names = F, quote = F)
```

4.4.2 ABCD Study

```
#Exclude those with medication use
##Make list of imputed datasets to feed to lavaan
implist_abcd <- lapply(seq(dsImp_abcd$m), function(im) complete(dsImp_abcd, im))

##Create dummy variables for categorical variables
for(x in seq(dsImp_abcd$m)){
  #Sex
  implist_abcd[[x]]$sex <-
    ifelse(implist_abcd[[x]]$sex_baseline_year_1_arm_1 == "F", 1, 0)
  #Maternal education
  implist_abcd[[x]]$maternal_education_high <-
    ifelse(implist_abcd[[x]]$maternal_education == "high", 1, 0)
  implist_abcd[[x]]$maternal_education_middle <-
    ifelse(implist_abcd[[x]]$maternal_education == "middle", 1, 0)
  #Child national origin
  implist_abcd[[x]]$child_nationalorigin_white <-
    ifelse(implist_abcd[[x]]$ethnicity == "White", 1, 0)
  implist_abcd[[x]]$child_nationalorigin_black <-
    ifelse(implist_abcd[[x]]$ethnicity == "Black", 1, 0)
  implist_abcd[[x]]$child_nationalorigin_hispanic <-
    ifelse(implist_abcd[[x]]$ethnicity == "Hispanic", 1, 0)
  implist_abcd[[x]]$child_nationalorigin_asian <-
    ifelse(implist_abcd[[x]]$ethnicity == "Asian", 1, 0)
  #Scanner site
  implist_abcd[[x]]$site1 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site01", 1, 0)
  implist_abcd[[x]]$site2 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site02", 1, 0)
  implist_abcd[[x]]$site3 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site03", 1, 0)
  implist_abcd[[x]]$site4 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site04", 1, 0)
  implist_abcd[[x]]$site5 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site05", 1, 0)
  implist_abcd[[x]]$site6 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site06", 1, 0)
  implist_abcd[[x]]$site7 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site07", 1, 0)
```

```
  implist_abcd[[x]]$site8 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site08", 1, 0)
  implist_abcd[[x]]$site9 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site09", 1, 0)
  implist_abcd[[x]]$site10 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site10", 1, 0)
  implist_abcd[[x]]$site11 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site11", 1, 0)
  implist_abcd[[x]]$site12 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site12", 1, 0)
  implist_abcd[[x]]$site13 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site13", 1, 0)
  implist_abcd[[x]]$site14 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site14", 1, 0)
  implist_abcd[[x]]$site15 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site15", 1, 0)
  implist_abcd[[x]]$site16 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site16", 1, 0)
  implist_abcd[[x]]$site17 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site17", 1, 0)
  implist_abcd[[x]]$site18 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site18", 1, 0)
  implist_abcd[[x]]$site19 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site19", 1, 0)
  implist_abcd[[x]]$site20 <-
    ifelse(implist_abcd[[x]]$site_id_l_baseline_year_1_arm_1 == "site20", 1, 0)
}

#Split T1-weighted sample
#Do this so that there are no missings in each dataframe
implist_t1w_abcd <- list()

for(x in seq(dsImp_abcd$m)){
  implist_t1w_abcd[[x]] <- subset(implist_abcd[[x]],
                                  !is.na(rowSums(implist_abcd[[x]][,18:105])) &
                                    psychotropic == 0)
}

#Create imputationlist object
data_abcd_t1w <- imputationList(implist_t1w_abcd)

#Create empty dataframes to store results
results_CLPM_abcd_sens4_m3_t1w <- data.frame()
fitmeaures_CLPM_abcd_sens4_m3_t1w <- data.frame()

#Specify rowcount to keep track of where we are in the loop
rowcount_clpm <- 1
rowcount_fit <- 1

#Specify the model for all brain morphology measures
for(x in sig_regions_abcd_clpm_t1w){
  CLPM_abcd <- paste0(
    '
```

```
# Estimate the lagged effects between the variables
  sum_dp_t2_sqrt_scaled +', x,"_2_year_follow_up_y_arm_1_scaled", ' ~
  sum_dp_t1_sqrt_scaled +', x,"_baseline_year_1_arm_1_scaled",

  #Estimate time independent predictors
  ' \n \n sum_dp_t1_sqrt_scaled ~ sex +  maternal_education_middle +
  maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
  child_nationalorigin_hispanic + child_nationalorigin_asian \n ',
  x,"_baseline_year_1_arm_1_scaled", ' ~ sex + maternal_education_middle +
  maternal_education_high + child_nationalorigin_white + child_nationalorigin_black +
  child_nationalorigin_hispanic + child_nationalorigin_asian +
  site1 + site2 + site3 + site4 + site5 + site6 + site7 + site 8 + site9 + site10 +
  site11 + site12 + site13 + site14 + site15 + site16 + site17 + site18 + site19 +
  site20
  \n \n

  #Estimate time dependent predictors
  sum_dp_t1_sqrt_scaled ~ interview_age_baseline_year_1_arm_1 \n
  sum_dp_t2_sqrt_scaled ~ interview_age_2_year_follow_up_y_arm_1 \n ',
  x,"_baseline_year_1_arm_1_scaled", ' ~ interview_age_baseline_year_1_arm_1 + handedness +
  smri_vol_scs_intracranialv_baseline_year_1_arm_1_scaled \n ',
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~ interview_age_2_year_follow_up_y_arm_1 + handedness +
  smri_vol_scs_intracranialv_2_year_follow_up_y_arm_1_scaled \n

  # Estimate the covariance between variables at the first wave.
  sum_dp_t1_sqrt_scaled ~~ ', x,"_baseline_year_1_arm_1_scaled", # Covariance

  # Estimate the covariances between the residuals of variables
  ' \n sum_dp_t2_sqrt_scaled ~~ ', x,"_2_year_follow_up_y_arm_1_scaled",

  # Estimate the (residual) variance of variables of interest
  ' \n sum_dp_t1_sqrt_scaled ~~ sum_dp_t1_sqrt_scaled \n ', # Variances
  x,"_baseline_year_1_arm_1_scaled", ' ~~ ', x,"_baseline_year_1_arm_1_scaled",
  ' \n sum_dp_t2_sqrt_scaled ~~ sum_dp_t2_sqrt_scaled \n ', # Residual variances
  x,"_2_year_follow_up_y_arm_1_scaled", ' ~~ ', x,"_2_year_follow_up_y_arm_1_scaled"
)

#Specify survey design to run clustered analyses (cluster = family ID)
survey_design_abcd = svydesign(id = ~rel_family_id_baseline_year_1_arm_1, data = data_abcd_t1w)

#Fit the model specified above
CLPM_abcd_sens4_fit <- lavaan(CLPM_abcd, data = implist_t1w_abcd[[1]], estimator = 'MLM')
#Perform complex sampling analysis
CLPM_abcd_sens4_CSA <- lavaan.survey(lavaan.fit = CLPM_abcd_sens4_fit,
                            survey.design = survey_design_abcd)
#Store coefficients of interest
#Which brain region
results_CLPM_abcd_sens4_m3_t1w[(rowcount_clpm+2),1] <- x
#Cross-sectional associations T1
results_CLPM_abcd_sens4_m3_t1w[c((rowcount_clpm+2)),2:4] <-
  summary(CLPM_abcd_sens4_CSA, fit.measures = T, standardized = T)[[2]][c(48),c(5,6,8)]
#Autoregressive parameters
#CBCL
```

```
  results_CLPM_abcd_sens4_m3_t1w[c((rowcount_clpm+2)),5:7] <-
    summary(CLPM_abcd_sens4_CSA, fit.measures = T, standardized = T)[[2]][1,c(5,6,8)]
  #MRI
  results_CLPM_abcd_sens4_m3_t1w[c((rowcount_clpm+2)),8:10] <-
    summary(CLPM_abcd_sens4_CSA, fit.measures = T, standardized = T)[[2]][4,c(5,6,8)]
  #Cross-lagged parameters
  #CBCL -> MRI
  results_CLPM_abcd_sens4_m3_t1w[c((rowcount_clpm+2)),11:13] <-
    summary(CLPM_abcd_sens4_CSA, fit.measures = T, standardized = T)[[2]][3,c(5,6,8)]
  #MRI -> CBCL
  results_CLPM_abcd_sens4_m3_t1w[c((rowcount_clpm+2)),14:16] <-
    summary(CLPM_abcd_sens4_CSA, fit.measures = T, standardized = T)[[2]][2,c(5,6,8)]
  #Which measure (actually only needs to be specified once)
  results_CLPM_abcd_sens4_m3_t1w[1,c(3,6,9,12,15)] <-
    c("Cross-sectional", "CBCL", "MRI", "CBCL -> MRI", "MRI -> CBCL")
  results_CLPM_abcd_sens4_m3_t1w[2,] <- c("Brain region", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value",
                              "B", "S.E.", "p-value", "B", "S.E.", "p-value")
  #Store summary statistics
  fitmeaures_CLPM_abcd_sens4_m3_t1w[rowcount_fit, 1:5] <-
    c(x, summary(CLPM_abcd_sens4_CSA, fit.measures = T,
                 standardized = T)[[1]][c("cfi.robust", "tli.robust", "rmsea.robust", "srmr")])
  #Adapt rowcounts to make sure results are stored properly
  rowcount_clpm <- rowcount_clpm + 1
  rowcount_fit <- rowcount_fit + 1
}

#Store output in CSV files
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_CLPM_abcd_sens4_m3_t1w, "results_CLPM_abcd_sens4_m3_t1w.csv",
          row.names = F, quote = F)
write.csv(fitmeaures_CLPM_abcd_sens4_m3_t1w, "fitmeaures_CLPM_abcd_sens4_m3_t1w.csv",
          row.names = F, quote = F)
```

4.5 Meta-analysis

In our final sensitivity analysis we combine the results obtained in Generation R and the ABCD study and meta-analyze the results. To accommodate this, we use the CLPM in Generation R using the data from T2 and T3 only.

```
#Create dataframe with individual estimates & SE's for both samples
#This is done separately for each arrow in the model

#T1-weighted data
#Model 3
results_meta_CS_m3_t1w <-
  data.frame(results_CLPM_2tp_genr_m3_t1w[3:45,1:4],
             results_CLPM_abcd_m3_t1w[3:45,1:4])

results_meta_AR_CBCL_m3_t1w <-
  data.frame(results_CLPM_2tp_genr_m3_t1w[3:45,c(1,5:7)],
             results_CLPM_abcd_m3_t1w[3:45,c(1,5:7)])

results_meta_AR_MRI_m3_t1w <-
```

```
  data.frame(results_CLPM_2tp_genr_m3_t1w[3:45,c(1,8:10)],
             results_CLPM_abcd_m3_t1w[3:45,c(1,8:10)])

results_meta_CL_CBCL_to_MRI_m3_t1w <-
  data.frame(results_CLPM_2tp_genr_m3_t1w[3:45,c(1,11:13)],
             results_CLPM_abcd_m3_t1w[3:45,c(1,11:13)])

results_meta_CL_MRI_to_CBCL_m3_t1w <-
  data.frame(results_CLPM_2tp_genr_m3_t1w[3:45,c(1,14:16)],
             results_CLPM_abcd_m3_t1w[3:45,c(1,14:16)])

#DTI data
#Model 2
results_meta_CS_m2_dti <-
  data.frame(results_CLPM_2tp_genr_m2_dti[3:16,1:4],
             results_CLPM_abcd_m2_dti[c(3:7,13,15,8:12,14,16),1:4])

results_meta_AR_CBCL_m2_dti <-
  data.frame(results_CLPM_2tp_genr_m2_dti[3:16,c(1,5:7)],
             results_CLPM_abcd_m2_dti[c(3:7,13,15,8:12,14,16),c(1,5:7)])

results_meta_AR_MRI_m2_dti <-
  data.frame(results_CLPM_2tp_genr_m2_dti[3:16,c(1,8:10)],
             results_CLPM_abcd_m2_dti[c(3:7,13,15,8:12,14,16),c(1,8:10)])

results_meta_CL_CBCL_to_MRI_m2_dti <-
  data.frame(results_CLPM_2tp_genr_m2_dti[3:16,c(1,11:13)],
             results_CLPM_abcd_m2_dti[c(3:7,13,15,8:12,14,16),c(1,11:13)])

results_meta_CL_MRI_to_CBCL_m2_dti <-
  data.frame(results_CLPM_2tp_genr_m2_dti[3:16,c(1,14:16)],
             results_CLPM_abcd_m2_dti[c(3:7,13,15,8:12,14,16),c(1,14:16)])

#Specify the n for each study
n <- c(nrow(df_final_genr), nrow(df_final_abcd))

#Specify sample names
sample_names <- c("GenR", "ABCD")

#Create temporary empty dataframe to store all results
results_meta <- list()

#Specify arrows and models to meta-analyze
arrows_models_names <-
  c("results_meta_CS_m3_t1w", "results_meta_AR_CBCL_m3_t1w",
    "results_meta_AR_MRI_m3_t1w", "results_meta_CL_CBCL_to_MRI_m3_t1w",
    "results_meta_CL_MRI_to_CBCL_m3_t1w",
    "results_meta_CS_m2_dti", "results_meta_AR_CBCL_m2_dti",
    "results_meta_AR_MRI_m2_dti", "results_meta_CL_CBCL_to_MRI_m2_dti",
    "results_meta_CL_MRI_to_CBCL_m2_dti")

#Run meta-analyses
#Loop over all models and arrows
```

```r
for(x in arrows_models_names){
  newdf <- get(x)
  #Rename columns to make them consistent
  colnames(newdf) <-
    c("names_genr", "est_genr", "se_genr", "pval_genr",
      "names_abcd", "est_abcd", "se_abcd", "pval_abcd")
  #Loop over all structures to get results
  for (y in 1:nrow(newdf)){
    #Store estimates for one path
    estimates <- as.numeric(c(newdf$est_genr[y], newdf$est_abcd[y]))
    #Same for standard errors
    se <- as.numeric(c(newdf$se_genr[y], newdf$se_abcd[y]))
    #Run a meta-analysis on those estimates & standard errors
    temp <- metagen(estimates, se, studlab = sample_names, n.e = n, fixed = T)
    #Save results
    newdf[y,"est_meta"] <-  temp$TE.fixed
    newdf[y,"se_meta"] <- temp$seTE.fixed
    newdf[y,"pval_meta"] <- temp$pval.fixed
  }
  #Store all results in list
  results_meta[[x]] <- newdf
}

#Store results in table
varnames <- c(results_CLPM_2tp_genr_m3_t1w[3:45,1],results_CLPM_2tp_genr_m2_dti[3:16,1])
results_meta_fin <-
  #Cross-sectional
  cbind(varnames, rbind(cbind(round(results_meta[[1]][,c("est_meta", "se_meta", "pval_meta")],3),
              #Autoregressive CBCL
              round(results_meta[[2]][,c("est_meta", "se_meta", "pval_meta")],3),
              #Autoregressive MRI
              round(results_meta[[3]][,c("est_meta", "se_meta", "pval_meta")],3),
              #Cross-lagged CBCL to MRI
              round(results_meta[[4]][,c("est_meta", "se_meta", "pval_meta")],3),
              #Cross-lagged MRI to CBCL
              round(results_meta[[5]][,c("est_meta", "se_meta", "pval_meta")],3)),
        cbind(round(results_meta[[6]][,c("est_meta", "se_meta", "pval_meta")],3),
              #Autoregressive CBCL
              round(results_meta[[7]][,c("est_meta", "se_meta", "pval_meta")],3),
              #Autoregressive MRI
              round(results_meta[[8]][,c("est_meta", "se_meta", "pval_meta")],3),
              #Cross-lagged CBCL to MRI
              round(results_meta[[9]][,c("est_meta", "se_meta", "pval_meta")],3),
              #Cross-lagged MRI to CBCL
              round(results_meta[[10]][,c("est_meta", "se_meta", "pval_meta")],3))
  ))

#Print results
results_meta_fin %>%
  kable(digits = 2,
        format = "latex",
      caption="Results meta-analyses",
      #col.names = c("n", "(M, SD)/%/(Median, IQR)"),
```

```
      align="r") %>%
  kable_classic(full_width = F, html_font = "helvetica")

#Save results
setwd("V:/medewerkers/074008 Blok, E/Dysregulation/2.Bidirectional/4.Results")
write.csv(results_meta_fin, "results_meta_fin.csv",
          row.names = F, quote = F)
```