

Data analysis script 2: Testing for replication of ERP to DDM concordance in RDoC dataset

Template Rmd

Brent Rappaport

2023-05-23

Contents

About	2
Get Setup	2
Clear everything & set width	2
Load Libraries	2
Function correlation_matrix	2
Load Data	4
Remove subjects with bad data	5
Make residualized score	5
Remove outliers	5
Standardize scores	6
Multiple regressions	8
Convergence with neuropsych measures	12
Inhibition & DDM	12
Set-shifting & DDM	15
Heritability	20
Split half reliability (Spearman-Brown prophecy)	20
Correlations	27
ERN amplitude, Neuropsych, & DDM	27
SOBP Stats	27
Save datasets	29

About

This script does preliminary data analysis comparing the psychometrics of HDDM models vs. raw accuracy vs. NIH Toolbox derived score for the Flanker task of the RDoC study.

v = **drift rate**: “The parameter of primary interest for the present study is the drift rate, v , which is the average rate of approach to a boundary and indexes the quality or strength of evidence extracted from the stimulus. A large value of drift indicates strong decision evidence, meaning the decision process will approach the appropriate boundary quickly, leading to fast and accurate responses.” Larger values of v mean faster accumulation of evidence (faster rate), larger t means slower non-decision time processing.

z = **response bias**: “If participants were biased toward one of the two responses (e.g., by increasing the proportion of one response over the other), they would move their starting point closer to that boundary. This produces faster and more probable responses at that boundary since less evidence is needed to reach it.” Distance between the the start of the drift process and upper boundary.

a = **separation between the two boundaries**: “response caution or speed/accuracy tradeoffs. If the boundary separation is relatively small, responses will take less time to reach a boundary, leading to faster responses, but they will also be more likely to reach the wrong boundary due to noise in the process, leading to more errors.” Larger a means more separation between the correct and incorrect response boundaries

t = **non-decision time**: “takes into account the duration of nondecisional processes. . . such processes may comprise basic encoding processes, the configuration of working memory for a task, and processes of response execution (i.e., motor activity).” (Voss et al., 2013)

Per Allie, larger values of v mean faster accumulation of evidence (faster rate), larger t means slower non-decision time processing, larger a means more separation between the correct and incorrect response boundaries, and z is the distance from the upper boundary to the start of the drift process.

Get Setup

Clear everything & set width

Load Libraries

```
## [1] "/Users/brentrappaport/Documents/temp_files/DDM/work"
```

Function correlation_matrix

```
correlation_matrix <- function(df,
                                type = "pearson",
                                digits = 3,
                                decimal.mark = ".",
                                use = "all",
                                show_significance = TRUE,
                                replace_diagonal = FALSE,
                                replacement = ""){

  # check arguments
  stopifnot({
    is.numeric(digits)
    digits >= 0
  })
```

```

    use %in% c("all", "upper", "lower")
    is.logical(replace_diagonal)
    is.logical(show_significance)
    is.character(replacement)
  })
  # we need the Hmisc package for this
  require(Hmisc)

  # retain only numeric and boolean columns
  isNumericOrBoolean = vapply(df, function(x) is.numeric(x) | is.logical(x), logical(1))
  if (sum(!isNumericOrBoolean) > 0) {
    cat('Dropping non-numeric/-boolean column(s):', paste(names(isNumericOrBoolean)[!isNumericOrBoolean])
  }
  df = df[isNumericOrBoolean]

  # transform input data frame to matrix
  x <- as.matrix(df)

  # run correlation analysis using Hmisc package
  correlation_matrix <- Hmisc::rcorr(x, type = )
  R <- correlation_matrix$r # Matrix of correlation coefficients
  p <- correlation_matrix$p # Matrix of p-value

  # transform correlations to specific character format
  Rformatted = formatC(R, format = 'f', digits = digits, decimal.mark = decimal.mark)

  # if there are any negative numbers, we want to put a space before the positives to align all
  if (sum(R < 0) > 0) {
    Rformatted = ifelse(R > 0, paste0(' ', Rformatted), Rformatted)
  }

  # add significance levels if desired
  if (show_significance) {
    # define notions for significance levels; spacing is important.
    stars <- ifelse(is.na(p), " ", ifelse(p < .001, "***", ifelse(p < .01, "** ", ifelse(p < .05, "*
    Rformatted = paste0(Rformatted, stars)
  }

  # build a new matrix that includes the formatted correlations and their significance stars
  Rnew <- matrix(Rformatted, ncol = ncol(x))
  rownames(Rnew) <- colnames(x)
  colnames(Rnew) <- paste(colnames(x), "", sep = " ")

  # replace undesired values
  if (use == 'upper') {
    Rnew[lower.tri(Rnew, diag = replace_diagonal)] <- replacement
  } else if (use == 'lower') {
    Rnew[upper.tri(Rnew, diag = replace_diagonal)] <- replacement
  } else if (replace_diagonal) {
    diag(Rnew) <- replacement
  }

  return(Rnew)
}

```

```
save_correlation_matrix = function(df, filename, ...) {
  write.csv2(correlation_matrix(df, ...), file = filename)
}
```

Load Data

Remember to immediately rename and remove. Avoid overwriting old data.

```
here::i_am("work/analysis/do01_LDDM_rdoc.Rmd")
```

```
## here() starts at /Users/brentrappaport/Documents/temp_files/DDM
```

```
LDDM_do1_alt_rdoc <- read.csv(here("./RDoC/DDM_Results/Block-Based/RDoC_Day1_Block11_Alternative_Models"))
```

```
LDDM_do1_rdoc <- read_sav(here("./work/data/RDoC_ERN_0-80ms_Final.sav"))
LDDM_do1_rdoc$ID <- LDDM_do1_rdoc$SubjectID
```

```
LDDM_do2_rdoc <- LDDM_do1_rdoc %>%
  left_join(LDDM_do1_alt_rdoc, by="ID") %>%
  filter(DQ >= 0.5)
```

```
load(file=here("./work/data/LDDM_cleaning04_fullbeh_rdoc.RData"))
```

```
load(file=here("./work/data/LDDM_cleaning04_rdoc_calc4.RData"))
```

```
## Load Color-Word D-KEFS test (Stroop) data
```

```
LDDM_do1_rdoc_iq <- read_sav(here("./work/data/RDoC_WTAR_FINAL_1.28.2019.sav"))
LDDM_do1_rdoc_iq$ID <- LDDM_do1_rdoc_iq$SubjectID
```

```
# lapply(LDDM_do1_rdoc_iq, attr, "label")
```

```
LDDM_do1_neuro <- read_sav(here("./work/data/RDoC_Neuropsych_Cleaning_HL_FINAL_3.19.18.sav"))
LDDM_do1_neuro$ID <- LDDM_do1_neuro$SubjectID
```

```
# LDDM_do1_education <- read_sav(here("./work/data/FINAL_RDoC_Proband_Sibling_ALL_Questionnaires_ItemLevel.sav"))
```

```
## Load data on participant's gender
```

```
LDDM_do3_rdoc_gender <- readxl::read_excel(here("./work/data/RDoC_ProbandSiblingCombined All 12.3.xlsx"))
LDDM_do3_rdoc_gender$ID <- LDDM_do3_rdoc_gender$`SubjectID of RDoC_ProbandSibling_AllQuestionnaires_Proband`
```

```
LDDM_do3_rdoc_gender$Gender <- LDDM_do3_rdoc_gender$`Gender of RDoC_ProbandSiblingCombined 12.3.jmp`
```

```
LDDM_do3_rdoc_gender_only <- LDDM_do3_rdoc_gender %>%
  select(ID, Gender)
```

```
LDDM_do2_neuro <- LDDM_do1_neuro %>%
  left_join(LDDM_do1_rdoc_iq, by="ID")
```

```
LDDM_do3_rdoc <- LDDM_do2_rdoc %>%
  left_join(LDDM_do3_rdoc_gender_only, by="ID") %>%
  left_join(LDDM_cleaning04_rdoc_calc4, by="ID") %>%
  left_join(LDDM_do2_neuro, by="ID") %>%
  select(-starts_with("Reason"))
```

```
# LDDM_do3_rdoc_no_outliers <- LDDM_do2_rdoc_no_outliers %>%
# left_join(LDDM_cleaning04_rdoc_calc4, by="ID")
#
```

Remove subjects with bad data

```
all_rdoc <- read_sav(here("./RDoC/All_Enrolled_IDs.sav"))
all_ern_rdoc<- read_sav(here("./RDoC/RDoC_ERN_0-80ms_Final.sav"))

all_rdoc <- all_rdoc %>%
  mutate(Eligible_use_this = Eligible) %>%
  select(SubjectID, Eligible_use_this, Reason)

all_ern_rdoc <- all_ern_rdoc %>%
  mutate(Has_atleast_10_error_trials = all_ern_rdoc$Number_of_errors_included_in_average>9,
         good_data = all_ern_rdoc$DQ>=0.5 & all_ern_rdoc$DQ<2) %>%
  select(SubjectID, Has_atleast_10_error_trials, good_data)

LDDM_do3_rdoc_with_bad_data <- full_join(LDDM_do3_rdoc, all_rdoc, by="SubjectID") %>%
  left_join(all_ern_rdoc, by="SubjectID")

LDDM_do3_rdoc <- full_join(LDDM_do3_rdoc, all_rdoc, by="SubjectID") %>%
  full_join(all_ern_rdoc, by="SubjectID") %>%
  filter(Eligible_use_this==1) %>%
  filter(Has_atleast_10_error_trials==TRUE) %>%
  filter(good_data==TRUE)
```

Make residualized score

```
electrodes_list <- c("Fz","Cz","FCz")

# Calculate residualized ERN and difference score ERN (for verification of direction)
for (ex in electrodes_list[1:3]){
  eval(parse(text=paste0('LDDM_do3_rdoc$',ex,'_ERN_080 <- stdres(lm(',ex,'3chan_error_500_300 ~ ',ex,
  eval(parse(text=paste0('LDDM_do3_rdoc$',ex,'_ERN_080_diff <- LDDM_do3_rdoc$',ex,'3chan_error_500_300
  eval(parse(text=paste0('LDDM_do3_rdoc$',ex,'_ERN_080_diff <- as.numeric(LDDM_do3_rdoc$',ex,'_ERN_080
  eval(parse(text=paste0('LDDM_do3_rdoc$',ex,'_ERN_080 <- as.numeric(LDDM_do3_rdoc$',ex,'_ERN_080'))))
}

LDDM_do3_rdoc$B11_avtz_D0_v_v <- rowMeans(LDDM_do3_rdoc[,c('B11_avtz_D0_v_v_con','B11_avtz_D0_v_v_incon
LDDM_do3_rdoc$B11_avt_D0_v_v <- rowMeans(LDDM_do3_rdoc[,c('B11_avt_D0_v_v_con','B11_avt_D0_v_v_incon')]
LDDM_do3_rdoc$B11_avtz_D0_v_v <- rowMeans(LDDM_do3_rdoc[,c('B11_avtz_D0_v_v_con','B11_avtz_D0_v_v_incon
```

Remove outliers

```
# lapply(LDDM_do3_rdoc, attr, "label")
LDDM_do3_rdoc$NL_time_rev <- -1*LDDM_do3_rdoc$NL_time
LDDM_do3_rdoc$InSw_time_rev <- -1*LDDM_do3_rdoc$InSw_time

LDDM_do3_rdoc$exec_composite <- rowMeans(LDDM_do3_rdoc[,c("FF_cor","Switch_cor","NL_time_rev","InSw_tim
```

```

LDDM_do3_rdoc_no_outliers <- LDDM_do3_rdoc %>%
  mutate(FCz_ERN_080 = Winsorize(FCz_ERN_080,
                                minval=mean(FCz_ERN_080, na.rm=T)-(3*sd(FCz_ERN_080, na.rm=T)),
                                maxval=mean(FCz_ERN_080, na.rm=T)+(3*sd(FCz_ERN_080, na.rm=T)),
                                na.rm=T),
    Inhib_time = Winsorize(Inhib_time,
                           minval=mean(Inhib_time, na.rm=T)-(3*sd(Inhib_time, na.rm=T)),
                           maxval=mean(Inhib_time, na.rm=T)+(3*sd(Inhib_time, na.rm=T)),
                           na.rm=T),
    exec_composite = Winsorize(exec_composite,
                               minval=mean(exec_composite, na.rm=T)-(3*sd(exec_composite, na.rm=T)),
                               maxval=mean(exec_composite, na.rm=T)+(3*sd(exec_composite, na.rm=T)),
                               na.rm=T)) %>%
  mutate(FCz_ERN_080 = FCz_ERN_080*-1)

```

Standardize scores

```

var_list <- c("B11_avtz_D0_v_v", "B11_avtz_D0_v_v_con", "B11_avtz_D0_v_v_incon", "B11_avtz_D0_v_a", "B11_avtz_D0_v_t",
             "B11_avt_v", "B11_avt_a", "B11_avt_t",
             "B11_avtz_v", "B11_avtz_a", "B11_avtz_t", "B11_avtz_z",
             "B11_avt_D0_v_v", "B11_avt_D0_v_v_con", "B11_avt_D0_v_v_incon", "B11_avt_D0_v_a", "B11_avt_D0_v_t",
             "B11_avtz_D0_v_v", "B11_avtz_D0_v_v_con", "B11_avtz_D0_v_v_incon", "B11_avtz_D0_v_a", "B11_avtz_D0_v_t",
             "flanker_score_rdoc",
             "FCz_ERN_080", "FCz_ERN_080_diff",
             "Inhib_time", "exec_composite", "Pred_FSIQ", "Motor_time", "accuracy", "accuracy_congruent_log")

for (v in var_list){
  print(paste0("LDDM_do3_rdoc$", v))
  eval(parse(text=paste0('LDDM_do3_rdoc$', v, '_z <- scale(LDDM_do3_rdoc$', v, ', center=T, scale=T)'))))
}

```

```

## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v_con"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v_incon"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_a"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_t"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_z"
## [1] "LDDM_do3_rdoc$B11_avt_v"
## [1] "LDDM_do3_rdoc$B11_avt_a"
## [1] "LDDM_do3_rdoc$B11_avt_t"
## [1] "LDDM_do3_rdoc$B11_avtz_v"
## [1] "LDDM_do3_rdoc$B11_avtz_a"
## [1] "LDDM_do3_rdoc$B11_avtz_t"
## [1] "LDDM_do3_rdoc$B11_avtz_z"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_v"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_v_con"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_v_incon"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_a"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_t"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v_con"

```

```
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v_incon"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_a"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_t"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_z"
## [1] "LDDM_do3_rdoc$flanker_score_rdoc"
## [1] "LDDM_do3_rdoc$FCz_ERN_080"
## [1] "LDDM_do3_rdoc$FCz_ERN_080_diff"
## [1] "LDDM_do3_rdoc$Inhib_time"
## [1] "LDDM_do3_rdoc$exec_composite"
## [1] "LDDM_do3_rdoc$Pred_FSIQ"
## [1] "LDDM_do3_rdoc$Motor_time"
## [1] "LDDM_do3_rdoc$accuracy"
## [1] "LDDM_do3_rdoc$accuracy_congruent_log"
## [1] "LDDM_do3_rdoc$accuracy_incongruent"
```

```
for (v in var_list){
  print(paste0("LDDM_do3_rdoc$",v))
  eval(parse(text=paste0('LDDM_do3_rdoc_no_outliers$',v,'_z <- scale(LDDM_do3_rdoc_no_outliers$',v,', c
}
```

```
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v_con"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v_incon"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_a"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_t"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_z"
## [1] "LDDM_do3_rdoc$B11_avt_v"
## [1] "LDDM_do3_rdoc$B11_avt_a"
## [1] "LDDM_do3_rdoc$B11_avt_t"
## [1] "LDDM_do3_rdoc$B11_avtz_v"
## [1] "LDDM_do3_rdoc$B11_avtz_a"
## [1] "LDDM_do3_rdoc$B11_avtz_t"
## [1] "LDDM_do3_rdoc$B11_avtz_z"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_v"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_v_con"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_v_incon"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_a"
## [1] "LDDM_do3_rdoc$B11_avt_D0_v_t"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v_con"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_v_incon"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_a"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_t"
## [1] "LDDM_do3_rdoc$B11_avtz_D0_v_z"
## [1] "LDDM_do3_rdoc$flanker_score_rdoc"
## [1] "LDDM_do3_rdoc$FCz_ERN_080"
## [1] "LDDM_do3_rdoc$FCz_ERN_080_diff"
## [1] "LDDM_do3_rdoc$Inhib_time"
## [1] "LDDM_do3_rdoc$exec_composite"
## [1] "LDDM_do3_rdoc$Pred_FSIQ"
## [1] "LDDM_do3_rdoc$Motor_time"
## [1] "LDDM_do3_rdoc$accuracy"
## [1] "LDDM_do3_rdoc$accuracy_congruent_log"
## [1] "LDDM_do3_rdoc$accuracy_incongruent"
```

```
make_CI <- function(lower, upper) {
  paste0("[", round(lower,2), ", ", round(upper,2), "]")
}
```

Multiple regressions

```
i=1
mr_ERN_table_rdoc <- as.data.frame(matrix(nrow=6, ncol=4))
mr_ERN_table_rdoc[,1] <- c("Drift rate", "Drift rate (congruent)", "Drift rate (incongruent)", "Boundary s
mr_ERN_table_rdoc_raw <- as.data.frame(matrix(nrow=6, ncol=4))
mr_ERN_table_rdoc_raw[,1] <- c("Drift rate", "Drift rate (congruent)", "Drift rate (incongruent)", "Bounda

for (var in c("B11_avtz_DO_v_v_z", "B11_avtz_DO_v_v_con_z", "B11_avtz_DO_v_v_incon_z", "B11_avtz_DO_v_a_z"
  eval(parse(text=paste0("
    model <- lmer(FCz_ERN_080_z ~ ", var, " + flanker_score_rdoc_z + accuracy_incongruent_z + Ag

    mr_ERN_table_rdoc[i,2] <- paste0(round(tidy(model)$estimate[2],2), ifelse(tidy(model)$p.va
      make_CI(confint(model)[4,1], confint(model)[4,2]))
    mr_ERN_table_rdoc[i,3] <- paste0(round(tidy(model)$estimate[3],2), ifelse(tidy(model)$p.va
      make_CI(confint(model)[5,1], confint(model)[5,2]))
    mr_ERN_table_rdoc[i,4] <- paste0(round(tidy(model)$estimate[4],2), ifelse(tidy(model)$p.va
      make_CI(confint(model)[6,1], confint(model)[6,2]))

    mr_ERN_table_rdoc_raw[i,2] <- tidy(model)$estimate[2]
    mr_ERN_table_rdoc_raw[i,3] <- confint(model)[4,1]
    mr_ERN_table_rdoc_raw[i,4] <- confint(model)[4,2]

    mr_ERN_table_rdoc_raw[i,5] <- tidy(model)$estimate[3]
    mr_ERN_table_rdoc_raw[i,6] <- confint(model)[5,1]
    mr_ERN_table_rdoc_raw[i,7] <- confint(model)[5,2]

    mr_ERN_table_rdoc_raw[i,8] <- tidy(model)$estimate[4]
    mr_ERN_table_rdoc_raw[i,9] <- confint(model)[6,1]
    mr_ERN_table_rdoc_raw[i,10] <- confint(model)[6,2]
    ")))

  i=i+1
}
```

```
## Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
```


[illegible]

[illegible]

[illegible]

```

colnames(mr_ERN_table_rdoc) <- c("Parameters", "Drift rate", "NIH Toolbox", "Raw accuracy")
colnames(mr_ERN_table_rdoc_congruent) <- c("Parameters", "Drift rate", "NIH Toolbox", "Raw accuracy (congruent)")

write.csv(mr_ERN_table_rdoc, here("./work/tables/mr_ERN_table_rdoc.csv"))
write.csv(mr_ERN_table_rdoc_raw, here("./work/tables/mr_ERN_table_rdoc_raw.csv"))

write.csv(mr_ERN_table_rdoc_congruent, here("./work/tables/mr_ERN_table_rdoc_congruent.csv"))
write.csv(mr_ERN_table_rdoc_congruent_raw, here("./work/tables/mr_ERN_table_rdoc_congruent_raw.csv"))

```

Convergence with neuropsych measures

Inhibition & DDM

```

LDDM_do3_rdoc_no_outliers$NL_time_rev <- -1*LDDM_do3_rdoc_no_outliers$NL_time
LDDM_do3_rdoc_no_outliers$Inhib_time_rev <- -1*LDDM_do3_rdoc_no_outliers$Inhib_time
LDDM_do3_rdoc_no_outliers$Inhib_time_rev_z <- scale(LDDM_do3_rdoc_no_outliers$Inhib_time_rev, center=T,

i=1
mr_INHIB_table_rdoc_nocov <- as.data.frame(matrix(nrow=6, ncol=4))
mr_INHIB_table_rdoc_nocov[,1] <- c("Drift rate", "Drift rate (congruent)", "Drift rate (incongruent)", "Bound")

mr_INHIB_table_rdoc_cov <- as.data.frame(matrix(nrow=6, ncol=4))
mr_INHIB_table_rdoc_cov[,1] <- c("Drift rate", "Drift rate (congruent)", "Drift rate (incongruent)", "Bound")
mr_INHIB_table_rdoc_cov_raw <- as.data.frame(matrix(nrow=6, ncol=4))
mr_INHIB_table_rdoc_cov_raw[,1] <- c("Drift rate", "Drift rate (congruent)", "Drift rate (incongruent)", "Bound")

for (var in c("B11_avtz_DO_v_v_z", "B11_avtz_DO_v_v_con_z", "B11_avtz_DO_v_v_incon_z", "B11_avtz_DO_v_a_z")) {
  eval(parse(text=paste0("
    model <- lmer(Inhib_time_rev_z ~ ", var, " + flanker_score_rdoc_z + accuracy_incongruent_z +

    mr_INHIB_table_rdoc_nocov[i,2] <- paste0(round(tidy(model)$estimate[2], 2), ifelse(tidy(model)$estimate[2] < 0, "[-",
      make_CI(confint(model)[4,1], confint(model)[4,2]))
    mr_INHIB_table_rdoc_nocov[i,3] <- paste0(round(tidy(model)$estimate[3], 2), ifelse(tidy(model)$estimate[3] < 0, "[-",
      make_CI(confint(model)[5,1], confint(model)[5,2]))
    mr_INHIB_table_rdoc_nocov[i,4] <- paste0(round(tidy(model)$estimate[4], 2), ifelse(tidy(model)$estimate[4] < 0, "[-",
      make_CI(confint(model)[6,1], confint(model)[6,2]))

    model_cov <- lmer(Inhib_time_rev_z ~ ", var, " + flanker_score_rdoc_z + accuracy_incongruent_z +

    mr_INHIB_table_rdoc_cov[i,2] <- paste0(round(tidy(model_cov)$estimate[2], 2), ifelse(tidy(model_cov)$estimate[2] < 0, "[-",
      make_CI(confint(model_cov)[4,1], confint(model_cov)[4,2]))
    mr_INHIB_table_rdoc_cov[i,3] <- paste0(round(tidy(model_cov)$estimate[3], 2), ifelse(tidy(model_cov)$estimate[3] < 0, "[-",
      make_CI(confint(model_cov)[5,1], confint(model_cov)[5,2]))
    mr_INHIB_table_rdoc_cov[i,4] <- paste0(round(tidy(model_cov)$estimate[4], 2), ifelse(tidy(model_cov)$estimate[4] < 0, "[-",
      make_CI(confint(model_cov)[6,1], confint(model_cov)[6,2]))

    mr_INHIB_table_rdoc_cov_raw[i,2] <- tidy(model_cov)$estimate[2]
    mr_INHIB_table_rdoc_cov_raw[i,3] <- confint(model_cov)[4,1]
    mr_INHIB_table_rdoc_cov_raw[i,4] <- confint(model_cov)[4,2]
  ")
  i=i+1
}

```


[illegible]

```

## Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...
##Computing profile confidence intervals ...

colnames(mr_INHIB_table_rdoc_nocov) <- c("Parameters","Drift rate", "NIH Toolbox", "Raw accuracy (incongruent)", "Bound")
colnames(mr_INHIB_table_rdoc_cov) <- c("Parameters","Drift rate", "NIH Toolbox", "Raw accuracy (incongruent)", "Bound")

write.csv(mr_INHIB_table_rdoc_nocov, here("./work/tables/mr_INHIB_table_rdoc_nocov.csv"))
write.csv(mr_INHIB_table_rdoc_cov, here("./work/tables/mr_INHIB_table_rdoc_cov.csv"))
write.csv(mr_INHIB_table_rdoc_cov_raw, here("./work/tables/mr_INHIB_table_rdoc_cov_raw.csv"))

```

Set-shifting & DDM

```

# lapply(LDDM_do3_rdoc, attr, "label")
LDDM_do3_rdoc_no_outliers$NL_time_rev <- -1*LDDM_do3_rdoc_no_outliers$NL_time
LDDM_do3_rdoc_no_outliers$InSw_time_rev <- -1*LDDM_do3_rdoc_no_outliers$InSw_time

LDDM_do3_rdoc_no_outliers$exec_composite <- rowMeans(LDDM_do3_rdoc_no_outliers[,c("FF_cor","Switch_cor","Bound")])
LDDM_do3_rdoc_no_outliers$exec_composite_z <- scale(LDDM_do3_rdoc_no_outliers$exec_composite, center=T, scale=F)

i=1
mr_EXEC_table_rdoc_nocov <- as.data.frame(matrix(nrow=6, ncol=4))
mr_EXEC_table_rdoc_nocov[,1] <- c("Drift rate","Drift rate (congruent)","Drift rate (incongruent)","Bound")

mr_EXEC_table_rdoc_cov <- as.data.frame(matrix(nrow=6, ncol=4))
mr_EXEC_table_rdoc_cov[,1] <- c("Drift rate","Drift rate (congruent)","Drift rate (incongruent)","Bound")
mr_EXEC_table_rdoc_cov_raw <- as.data.frame(matrix(nrow=6, ncol=4))
mr_EXEC_table_rdoc_cov_raw[,1] <- c("Drift rate","Drift rate (congruent)","Drift rate (incongruent)","Bound")

for (var in c("B11_avtz_DO_v_v_z","B11_avtz_DO_v_v_con_z","B11_avtz_DO_v_v_incon_z","B11_avtz_DO_v_a_z")){
  eval(parse(text=paste0("
    model <- lmer(exec_composite_z ~ ",var," + flanker_score_rdoc_z + accuracy_incongruent_z + accuracy_congruent_z, data=LDDM_do3_rdoc_no_outliers)

    mr_EXEC_table_rdoc_nocov[i,2] <- paste0(round(tidy(model)$estimate[2],2), ifelse(tidy(model)$p.value[2]<0.001,"***",ifelse(tidy(model)$p.value[2]<0.01,"**",ifelse(tidy(model)$p.value[2]<0.05,"*",))))
    mr_EXEC_table_rdoc_nocov[i,3] <- paste0(round(tidy(model)$estimate[3],2), ifelse(tidy(model)$p.value[3]<0.001,"***",ifelse(tidy(model)$p.value[3]<0.01,"**",ifelse(tidy(model)$p.value[3]<0.05,"*",))))
    mr_EXEC_table_rdoc_nocov[i,4] <- paste0(round(tidy(model)$estimate[4],2), ifelse(tidy(model)$p.value[4]<0.001,"***",ifelse(tidy(model)$p.value[4]<0.01,"**",ifelse(tidy(model)$p.value[4]<0.05,"*",))))

    model_cov <- lmer(exec_composite_z ~ ",var," + flanker_score_rdoc_z + accuracy_z + Pred_FSI, data=LDDM_do3_rdoc_no_outliers)

    mr_EXEC_table_rdoc_cov[i,2] <- paste0(round(tidy(model_cov)$estimate[2],2), ifelse(tidy(model_cov)$p.value[2]<0.001,"***",ifelse(tidy(model_cov)$p.value[2]<0.01,"**",ifelse(tidy(model_cov)$p.value[2]<0.05,"*",))))
  })
}

```



```
## Warning in FUN(X[[i]], ...): non-monotonic profile for .sig01

## Warning in FUN(X[[i]], ...): bad spline fit for .sig01: falling back to linear
## interpolation

## Computing profile confidence intervals ...

## Warning in FUN(X[[i]], ...): non-monotonic profile for .sig01

## Warning in FUN(X[[i]], ...): bad spline fit for .sig01: falling back to linear
## interpolation

##Computing profile confidence intervals ...

##Warning in FUN(X[[i]], ...): non-monotonic profile for .sig01

##Warning in FUN(X[[i]], ...): bad spline fit for .sig01: falling back to linear
##interpolation

##Computing profile confidence intervals ...

##Warning in FUN(X[[i]], ...): non-monotonic profile for .sig01

##Warning in FUN(X[[i]], ...): bad spline fit for .sig01: falling back to linear
##interpolation

##Computing profile confidence intervals ...

##Warning in FUN(X[[i]], ...): non-monotonic profile for .sig01

##Warning in FUN(X[[i]], ...): bad spline fit for .sig01: falling back to linear
##interpolation

##Computing profile confidence intervals ...

##Warning in FUN(X[[i]], ...): non-monotonic profile for .sig01

##Warning in FUN(X[[i]], ...): bad spline fit for .sig01: falling back to linear
##interpolation

##Computing profile confidence intervals ...
```

[illegible]

Heritability

```
LDDM_do3_rdoc_sib <- LDDM_do3_rdoc %>%
  pivot_wider(id_cols=F_ID, names_from=S_ID,
              values_from=c(B11_avtz_D0_v_v_z, B11_avtz_D0_v_v_con_z, B11_avtz_D0_v_v_incon_z,
                            B11_avtz_D0_v_a_z, B11_avtz_D0_v_t_z, B11_avtz_D0_v_z_z, flanker_score_rdoc_z,

LDDM_heritability_table <- data.frame(parameters= c("v", "v_congruent", "v_incongruent", "a", "t", "z",
ICC=c(icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_v_z_1", "B11_avtz_D0_v_v_z_2"))],
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_v_con_z_1", "B11_avtz_D0_v_v_con_z_2")],
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_v_incon_z_1", "B11_avtz_D0_v_v_incon_z_2")],
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_a_z_1", "B11_avtz_D0_v_a_z_2")])$value,
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_t_z_1", "B11_avtz_D0_v_t_z_2")])$value,
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_z_z_1", "B11_avtz_D0_v_z_z_2")])$value,
icc(LDDM_do3_rdoc_sib[c("flanker_score_rdoc_z_1", "flanker_score_rdoc_z_2")]),
icc(LDDM_do3_rdoc_sib[c("accuracy_incongruent_z_1", "accuracy_incongruent_z_2")]),
icc(LDDM_do3_rdoc_sib[c("accuracy_z_1", "accuracy_z_2")])$value),

p=c(icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_v_z_1", "B11_avtz_D0_v_v_z_2")])$p,
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_v_con_z_1", "B11_avtz_D0_v_v_con_z_2")],
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_v_incon_z_1", "B11_avtz_D0_v_v_incon_z_2")],
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_a_z_1", "B11_avtz_D0_v_a_z_2")])$p.value,
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_t_z_1", "B11_avtz_D0_v_t_z_2")])$p.value,
icc(LDDM_do3_rdoc_sib[c("B11_avtz_D0_v_z_z_1", "B11_avtz_D0_v_z_z_2")])$p.value,
icc(LDDM_do3_rdoc_sib[c("flanker_score_rdoc_z_1", "flanker_score_rdoc_z_2")]),
icc(LDDM_do3_rdoc_sib[c("accuracy_incongruent_z_1", "accuracy_incongruent_z_2")]),
icc(LDDM_do3_rdoc_sib[c("accuracy_z_1", "accuracy_z_2")])$p.value))

write.csv(LDDM_heritability_table, here("./work/tables/rdoc_heritability_table.csv"))
```

Split half reliability (Spearman-Brown prophecy)

```
load(file=here("./work/data/LDDM_cleaning04_fullbeh_rdoc.RData"))

LDDM_cleaning04_rdoc_reliability <- LDDM_cleaning04_fullbeh_rdoc %>%
  group_by(subj_idx) %>%
  mutate(block = c(rep(1,30), rep(2,30), rep(3,30), rep(4,30), rep(5,30), rep(6,30), rep(7,30), rep(8,30), rep(9,30)))
  group_by(subj_idx) %>%
  mutate(trial = 1:330)

for (s in seq(15,166,15)){
  eval(parse(text=paste0('
LDDM_first',s,'_1 <- LDDM_cleaning04_rdoc_reliability %>% filter(trial<=',s,')
LDDM_second',s,'_1 <- LDDM_cleaning04_rdoc_reliability %>% filter(trial< ',(s*2)+1,' & trial>',s,')

LDDM_first',s,'_2 <- LDDM_first',s,'_1 %>%
  group_by(subj_idx) %>% # per subject
  summarise(accuracy_score = sum(response==1)*(5/length(trial))) # accuracy score per NIH Toolbox manual
LDDM_second',s,'_2 <- LDDM_second',s,'_1 %>%
  group_by(subj_idx) %>%
```

```

    summarise(accuracy_score = sum(response==1)*(5/length(trial))) # accuracy score per NIH Toolbox manual

LDDM_first',s,'_3 <- LDDM_first',s,'_1 %>%
  group_by(subj_idx) %>% # per subject
  filter(stim=="incongruent" & response==1) %>% # incongruent trials with correct response
  mutate(mean_rt = mean(rt),
          sd_rt = sd(rt)) %>% # compute individual mean and sd RT for use below
  filter(rt>=0.1 & rt>(mean_rt - 3*sd_rt) & rt<(mean_rt + 3*sd_rt)) %>% # remove trials less than 100ms
  summarise(med_rt = median(rt)*1000) %>% # compute individual level median RT
  mutate(rt_score = 5-(5*((log(med_rt)-log(250))/(log(1000)-log(250)))) # compute RT score to go into :
LDDM_second',s,'_3 <- LDDM_second',s,'_1 %>%
  group_by(subj_idx) %>% # per subject
  filter(stim=="incongruent" & response==1) %>% # incongruent trials with correct response
  mutate(mean_rt = mean(rt),
          sd_rt = sd(rt)) %>% # compute individual mean and sd RT for use below
  filter(rt>=0.1 & rt>(mean_rt - 3*sd_rt) & rt<(mean_rt + 3*sd_rt)) %>% # remove trials less than 100ms
  summarise(med_rt = median(rt)*1000) %>% # compute individual level median RT
  mutate(rt_score = 5-(5*((log(med_rt)-log(250))/(log(1000)-log(250)))) # compute RT score to go into :

LDDM_first',s,' <- LDDM_first',s,'_1 %>%
  full_join(LDDM_first',s,'_2, by = "subj_idx") %>%
  full_join(LDDM_first',s,'_3, by="subj_idx") %>%
  group_by(subj_idx) %>% # per subject
  mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
  summarise(flanker_score_list_rdoc = if_else(total_accuracy_perc>=0.8, accuracy_score+rt_score, accuracy_score,
          accuracy = mean(total_accuracy_perc)) %>% # if accuracy is above 80% then add accuracy and
  transmute(ID=subj_idx, flanker_score_list_rdoc=flanker_score_list_rdoc, accuracy=accuracy) %>%
  group_by(ID) %>% # per subject
  summarise(flanker_score_rdoc = mean(flanker_score_list_rdoc),
          accuracy = mean(accuracy))
LDDM_second',s,' <- LDDM_second',s,'_1 %>%
  full_join(LDDM_second',s,'_2, by = "subj_idx") %>%
  full_join(LDDM_second',s,'_3, by="subj_idx") %>%
  group_by(subj_idx) %>% # per subject
  mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
  summarise(flanker_score_list_rdoc = if_else(total_accuracy_perc>=0.8, accuracy_score+rt_score, accuracy_score,
          accuracy = mean(total_accuracy_perc)) %>% # if accuracy is above 80% then add accuracy and
  transmute(ID=subj_idx, flanker_score_list_rdoc=flanker_score_list_rdoc, accuracy=accuracy) %>%
  group_by(ID) %>% # per subject
  summarise(flanker_score_rdoc = mean(flanker_score_list_rdoc),
          accuracy = mean(accuracy))

LDDM_first_accuracy',s,' <- LDDM_first',s,'_1 %>%
  full_join(LDDM_first',s,'_2, by = "subj_idx") %>%
  full_join(LDDM_first',s,'_3, by="subj_idx") %>%
  group_by(subj_idx, stim) %>% # per subject and condition
  mutate(accuracy_by_stim = sum(response==1)/length(response)) %>%
  summarise(accuracy_by_stim=mean(accuracy_by_stim)) %>%
  pivot_wider(names_from=stim, values_from=accuracy_by_stim, id_cols=subj_idx, names_prefix="accuracy_")
LDDM_second_accuracy',s,' <- LDDM_second',s,'_1 %>%
  full_join(LDDM_second',s,'_2, by = "subj_idx") %>%
  full_join(LDDM_second',s,'_3, by="subj_idx") %>%
  group_by(subj_idx, stim) %>% # per subject and condition

```

```

mutate(accuracy_by_stim = sum(response==1)/length(response)) %>%
summarise(accuracy_by_stim=mean(accuracy_by_stim)) %>%
pivot_wider(names_from=stim, values_from=accuracy_by_stim, id_cols=subj_idx, names_prefix="accuracy_")

r_half_',s,' <- cor(LDDM_first',s,'$flanker_score_rdoc, LDDM_second',s,'$flanker_score_rdoc, method="spearmanr")
r_half_ci_',s,' <- ci_cor(LDDM_first',s,'$flanker_score_rdoc, LDDM_second',s,'$flanker_score_rdoc, method="spearmanr")
r_sb_cilower_',s,' <- (2*r_half_ci_',s,'$interval[1])/(1+r_half_ci_',s,'$interval[1])
r_sb_ciupper_',s,' <- (2*r_half_ci_',s,'$interval[2])/(1+r_half_ci_',s,'$interval[2])
r_sb_',s,' <- (2*r_half_',s,')/(1+r_half_',s,')

racc_half_',s,' <- cor(LDDM_first_accuracy',s,'$accuracy_incongruent, LDDM_second_accuracy',s,'$accuracy_incongruent, method="spearmanr")
racc_half_ci_',s,' <- ci_cor(LDDM_first_accuracy',s,'$accuracy_incongruent, LDDM_second_accuracy',s,'$accuracy_incongruent, method="spearmanr")
racc_sb_cilower_',s,' <- (2*racc_half_ci_',s,'$interval[1])/(1+racc_half_ci_',s,'$interval[1])
racc_sb_ciupper_',s,' <- (2*racc_half_ci_',s,'$interval[2])/(1+racc_half_ci_',s,'$interval[2])
racc_sb_',s,' <- (2*racc_half_',s,')/(1+racc_half_',s,')
'))))
}

```

```

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

```

```

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

```

```

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

```

```

## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.

```

```

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

```

```

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

```

```

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

```

```

## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

```

```

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

```



```

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.

## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.

```

```
## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.
```

```
## 'summarise()' has grouped output by 'subj_idx'. You can override using the
## '.groups' argument.
```

```
## Warning: Returning more (or less) than 1 row per 'summarise()' group was deprecated in dplyr 1.1.0.
## i Please use 'reframe()' instead.
## i When switching from 'summarise()' to 'reframe()', remember that 'reframe()' always returns an
##   ungrouped data frame and adjust accordingly.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.
```

```
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
## 'summarise()' has grouped output by 'subj_idx'. You can override using the '.groups' argument.
```

```
spearman_brown_rdoc <- data.frame(trials=seq(15,165,15),
                                rnih=rep(NA,length(seq(15,165,15))),
                                rnih_cilower=rep(NA,length(seq(15,165,15))),
                                rnih_ciupper=rep(NA,length(seq(15,165,15))),
                                racc=rep(NA,length(seq(15,165,15))),
                                racc_cilower=rep(NA,length(seq(15,165,15))),
                                racc_ciupper=rep(NA,length(seq(15,165,15))))

i=1
for (s in seq(15,165,15)){
  eval(parse(text=paste0('spearman_brown_rdoc[i,2] <- round(as.numeric(r_sb_',s, '),3)
                        spearman_brown_rdoc[i,3] <- round(as.numeric(r_sb_cilower_',s, '),3)
                        spearman_brown_rdoc[i,4] <- round(as.numeric(r_sb_ciupper_',s, '),3)
                        spearman_brown_rdoc[i,5] <- round(as.numeric(racc_sb_',s, '),3)
                        spearman_brown_rdoc[i,6] <- round(as.numeric(racc_sb_cilower_',s, '),3)
                        spearman_brown_rdoc[i,7] <- round(as.numeric(racc_sb_ciupper_',s, '),3)'))))
  i=i+1
}

sttr_rdpd_ddm_splithalf <- read.csv(here("./Split_Half/RDoC_splithalf.csv"))
spearman_brown_rdoc$rdriftcon <- sttr_rdpd_ddm_splithalf$avtz_D0_v_vcon
spearman_brown_rdoc$rdriftincon <- sttr_rdpd_ddm_splithalf$avtz_D0_v_vinc
spearman_brown_rdoc$rboundary_separation <- sttr_rdpd_ddm_splithalf$avtz_D0_v_a

# ggplot(spearman_brown_rdoc, aes(x=trials, y=r, group=1)) +
#   geom_line() +
#   geom_point() +
#   scale_y_continuous(limits = c(0, 1)) +
#   geom_ribbon(aes(ymin = r_ci_lower, ymax = r_ci_upper), alpha = 0.2)
```

Correlations

ERN amplitude, Neuropsych, & DDM

```
all_measures_rdoc <- c("FCz_ERN_080_z", "flanker_score_rdoc_z", "accuracy_z", "accuracy_incongruent_z", "ac
                        "B11_avtz_DO_v_v_z", "B11_avtz_DO_v_v_con_z", "B11_avtz_DO_v_v_incon_z", "B11_avtz_DO_v_v_

rdoc_correlations <- correlation_matrix(LDDM_do3_rdoc_no_outliers[c(all_measures_rdoc)], type = c("spea

## Loading required package: Hmisc

## Registered S3 methods overwritten by 'Hmisc':
##   method                from
##   [.labelled             expss
##   print.labelled         expss
##   as.data.frame.labelled expss

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:DescTools':
##
##   %nin%, Label, Mean, Quantile

## The following object is masked from 'package:psych':
##
##   describe

## The following objects are masked from 'package:dplyr':
##
##   src, summarize

## The following objects are masked from 'package:base':
##
##   format.pval, units

write.csv(rdoc_correlations, file=here("./work/tables/rdoc_correlations.csv"))

# ggplot(filter(LDDM_do3_rdoc, FCz_ERN_080 > -3 & FCz_ERN_080 < 3), aes(x=FCz_ERN_080_z, y=B11_avtz_DO_v_v_z,
#   geom_point() +
#   stat_smooth(method="lm")
```

SOBP Stats

```
sum(!is.na(LDDM_do3_rdoc_no_outliers$B11_avtz_DO_v_v_z))

## [1] 381
```

```
summary(lmer(FCz_ERN_080_z ~ B11_avtz_DO_v_v_con_z + flanker_score_rdoc_z + accuracy_z + (1 | F_ID), LD
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: FCz_ERN_080_z ~ B11_avtz_DO_v_v_con_z + flanker_score_rdoc_z +
## accuracy_z + (1 | F_ID)
## Data: LDDM_do3_rdoc_no_outliers
##
## REML criterion at convergence: 1074.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.0111 -0.6375  0.0039  0.5331  2.6529
##
## Random effects:
## Groups Name Variance Std.Dev.
## F_ID (Intercept) 0.03074 0.1753
## Residual 0.93613 0.9675
## Number of obs: 379, groups: F_ID, 230
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    0.01843    0.05113 207.50333   0.360 0.718859
## B11_avtz_DO_v_v_con_z 0.19612    0.05596 374.63952   3.505 0.000513 ***
## flanker_score_rdoc_z 0.02694    0.06319 371.73516   0.426 0.670117
## accuracy_z      0.04420    0.05854 374.92012   0.755 0.450746
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) B11__D fln___
## B11__DO_____ -0.002
## flnkr_scr__ 0.004 -0.379
## accuracy_z -0.005 0.017 -0.473
```

```
summary(lmer(FCz_ERN_080_z ~ B11_avtz_DO_v_v_incon_z + flanker_score_rdoc_z + accuracy_z + (1 | F_ID), l
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: FCz_ERN_080_z ~ B11_avtz_DO_v_v_incon_z + flanker_score_rdoc_z +
## accuracy_z + (1 | F_ID)
## Data: LDDM_do3_rdoc_no_outliers
##
## REML criterion at convergence: 1071
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.95015 -0.64827  0.01161  0.59062  2.42517
##
## Random effects:
## Groups Name Variance Std.Dev.
## F_ID (Intercept) 0.06428 0.2535
```

```

## Residual          0.89720  0.9472
## Number of obs: 379, groups: F_ID, 230
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    0.019715   0.051657 206.179519   0.382 0.703110
## B11_avtz_D0_v_v_incon_z  0.352207   0.090668 371.500485   3.885 0.000121 ***
## flanker_score_rdoc_z    -0.009006   0.065852 373.261448  -0.137 0.891294
## accuracy_z        -0.174432   0.080129 373.465427  -2.177 0.030116 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) B11__D fln___
## B11__D0_____ 0.001
## flnkr_scr__   0.003 -0.467
## accuracy_z    -0.004 -0.686 -0.004

summary(lmer(FCz_ERN_080_z ~ B11_avtz_D0_v_v_z + (1 | F_ID), LDDM_do3_rdoc_no_outliers))

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: FCz_ERN_080_z ~ B11_avtz_D0_v_v_z + (1 | F_ID)
## Data: LDDM_do3_rdoc_no_outliers
##
## REML criterion at convergence: 1063.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.02131 -0.63300  0.00945  0.57350  2.58669
##
## Random effects:
## Groups Name Variance Std.Dev.
## F_ID (Intercept) 0.03184  0.1784
## Residual 0.92044  0.9594
## Number of obs: 379, groups: F_ID, 230
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    0.01844   0.05078 208.67382   0.363 0.717
## B11_avtz_D0_v_v_z  0.24422   0.05027 375.29562  4.858 1.74e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr)
## B11_v_D0___ -0.001

```

Save datasets

```
save(LDDM_do3_rdoc_with_bad_data, file=here("./work/data/LDDM_do3_rdoc_with_bad_data.RData"))  
save(LDDM_do3_rdoc, file=here("./work/data/LDDM_do3_rdoc.RData"))  
save(LDDM_do3_rdoc_no_outliers, file=here("./work/data/LDDM_do3_rdoc_no_outliers.RData"))  
save(spearman_brown_rdoc, file=here("./work/data/spearman_brown_rdoc.RData"))
```