

Data prep script 2: Compute NIH TOolbox Flanker Score

Template Rmd

Brent Rappaport

2023-05-23

Contents

About	1
Get Setup	1
Clear everything & set width	1
Load Libraries	2
Get the Working Directory	2
Set seed	2
Load Data	2
Day 1	2
Only 20 trials	4
Day 2	4
Only 20 trials	5
Day 3	6
Only 20 trials	7
Closing out	8

About

This script computes the NIH Toolbox Flanker score

Get Setup

Clear everything & set width

```
options(width=80, Ncpus = 6) #Set width
rm(list=ls())               #Remove everything from environment
cat("\014")                 #Clear Console
```

Load Libraries

```
# renv::restore()      #restore environment
library(knitr)          #allows rmarkdown files
library(haven)          #helps import stata
library(questionr)      #allows lookfor function
library(MASS)           #calculate residualized scores
library(tidyverse)      #plotting/cleaning, etc.
library(broom)          #nice statistical output
library(here)           #nice file paths
library(expss)          #labeling variables/values
library(psych)          #used for statistical analyses
library(workflowr)      #helps with workflow
```

Get the Working Directory

```
here()
```

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

Set seed

```
set.seed(312)      #Set seed
```

Load Data

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

```
here::i_am("work/management/data02_LDDM_sttr.Rmd")
```

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

```
LDDM_cleaning04_fullbeh_d1 <- read.csv(here("DDM_Trialwise_Prepped/Block_Based/Day1/StTr_S1_ERN_HDDM_da
LDDM_cleaning04_fullbeh_d2 <- read.csv(here("DDM_Trialwise_Prepped/Block_Based/Day2/StTr_S2_ERN_HDDM_da
LDDM_cleaning04_fullbeh_d3 <- read.csv(here("DDM_Trialwise_Prepped/Block_Based/Day3/StTr_S3_ERN_HDDM_da

sttr_all_data <- read_sav(here("work/data/STTR_master_6.6.18_clean_ERN_PE_LPPbehav.sav")) %>%
  select(ID, Age)
```

Day 1

```

# Setup the dataset with the needed variables (eg., ID, block # 1-11, trial # 1-30 per block)
LDDM_cleaning04_d1_calc1 <- LDDM_cleaning04_fullbeh_d1 %>%
  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(
  group_by(subj_idx, block) %>%
  mutate(trial = 1:30) %>%
  filter(rt<999)

# Calculate the accuracy per subject (according to NIH toolbox this should be the number of correct responses)
LDDM_cleaning04_d1_calc2 <- LDDM_cleaning04_d1_calc1 %>%
  group_by(subj_idx) %>%
  reframe(accuracy_score = sum(response==1)*(5/length(trial)))

LDDM_cleaning04_d1_ind <- LDDM_cleaning04_d1_calc1 %>%
  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
  filter(rt>=0.1 & rt>(mean_rt - 3*sd_rt) & rt<(mean_rt + 3*sd_rt)) %>% #remove trials less than 100ms
  reframe(med_rt = median(rt)*1000) %>% # compute individual level median RT
  mutate(rt_score = 5-(5*((log(med_rt)-log(250))/(log(1000)-log(250)))),
         rt_score_truncated = 5-(5*((log(med_rt)-log(500))/(log(3000)-log(500)))))

LDDM_cleaning04_d1_calc3 <- LDDM_cleaning04_d1_calc1 %>%
  full_join(LDDM_cleaning04_d1_ind, by = 'subj_idx') %>%
  full_join(LDDM_cleaning04_d1_calc2, by='subj_idx') %>%
  group_by(subj_idx) %>% # per subject
  mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
  reframe(flanker_score_list_d1 = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score, accuracy_score),
         flanker_score_list_d1_truncated = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score_truncated, accuracy_score),
         accuracy = mean(total_accuracy_perc)) %>% # if accuracy is above 80% then add accuracy and rt score
  transmute(ID=subj_idx, flanker_score_list_d1=flanker_score_list_d1, flanker_score_list_d1_truncated=flanker_score_list_d1_truncated)
  group_by(ID) %>% # per subject
  reframe(flanker_score = mean(flanker_score_list_d1),
         flanker_score_truncated = mean(flanker_score_list_d1_truncated),
         accuracy = mean(accuracy))

LDDM_cleaning04_d1_accuracy_by_condition <- LDDM_cleaning04_d1_calc1 %>%
  full_join(LDDM_cleaning04_d1_ind, by = 'subj_idx') %>%
  full_join(LDDM_cleaning04_d1_calc2, by='subj_idx') %>%
  group_by(subj_idx, stim) %>% # per subject
  mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
  reframe(accuracy_by_stim = mean(total_accuracy_perc)) %>%
  pivot_wider(names_from=stim, values_from=accuracy_by_stim, id_cols=subj_idx, names_prefix="accuracy_")
  mutate(ID=subj_idx,
         accuracy_congruent_log = log(accuracy_congruent))

LDDM_cleaning04_d1_calc4 <- full_join(LDDM_cleaning04_d1_calc3, LDDM_cleaning04_d1_accuracy_by_condition)
select(-subj_idx)

```

Only 20 trials

```
# Setup the dataset with the needed variables (eg., ID, block # 1-11, trial # 1-30 per block)
LDDM_cleaning04_d1_calc1_short_incon <- LDDM_cleaning04_d1_calc1 %>%
  group_by(subj_idx) %>%
  filter(stim=="incongruent") %>%
  slice_head(n=8)

LDDM_cleaning04_d1_calc1_short_con <- LDDM_cleaning04_d1_calc1 %>%
  group_by(subj_idx) %>%
  filter(stim=="congruent") %>%
  slice_head(n=12)

LDDM_cleaning04_d1_calc1_short <- rbind(LDDM_cleaning04_d1_calc1_short_incon, LDDM_cleaning04_d1_calc1_short_con) %>%
  arrange(subj_idx)

# Calculate the accuracy per subject (according to NIH toolbox this should be the number of correct responses)
LDDM_cleaning04_d1_calc2_short <- LDDM_cleaning04_d1_calc1_short %>%
  group_by(subj_idx) %>%
  reframe(accuracy_score = sum(response==1)*(5/length(trial)))

LDDM_cleaning04_d1_ind_short <- LDDM_cleaning04_d1_calc1_short %>%
  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
  filter(rt>=0.1 & rt>(mean_rt - 3*sd_rt) & rt<(mean_rt + 3*sd_rt)) %>% #remove trials less than 100ms
  reframe(med_rt = median(rt)*1000) %>% # compute individual level median RT
  mutate(rt_score = 5-(5*((log(med_rt)-log(250))/(log(1000)-log(250)))),
         rt_score_truncated = 5-(5*((log(med_rt)-log(500))/(log(3000)-log(500)))))

LDDM_cleaning04_d1_calc3_short <- LDDM_cleaning04_d1_calc1_short %>%
  full_join(LDDM_cleaning04_d1_ind, by = 'subj_idx') %>%
  full_join(LDDM_cleaning04_d1_calc2, by='subj_idx') %>%
  group_by(subj_idx) %>% # per subject
  mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
  reframe(flanker_score_list_d1 = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score, accuracy_score),
         flanker_score_list_d1_truncated = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score_truncated, accuracy_score),
         accuracy = mean(total_accuracy_perc)) %>% # if accuracy is above 80% then add accuracy and rt score
  transmute(ID=subj_idx, flanker_score_list_d1=flanker_score_list_d1, flanker_score_list_d1_truncated=flanker_score_list_d1_truncated)
  group_by(ID) %>% # per subject
  reframe(flanker_score = mean(flanker_score_list_d1),
         flanker_score_truncated = mean(flanker_score_list_d1_truncated),
         accuracy = mean(accuracy))
```

Day 2

```
# Setup the dataset with the needed variables (eg., ID, block # 1-11, trial # 1-30 per block)
LDDM_cleaning04_d2_calc1 <- LDDM_cleaning04_fullbeh_d2 %>%
  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),rep(10,30)))
```

```

group_by(subj_idx, block) %>%
mutate(trial = 1:30) %>%
filter(rt<999)

# Calculate the accuracy per subject (according to NIH toolbox this should be the number of correct responses)
LDDM_cleaning04_d2_calc2 <- LDDM_cleaning04_d2_calc1 %>%
group_by(subj_idx) %>%
reframe(accuracy_score = sum(response==1)*(5/length(trial)))

LDDM_cleaning04_d2_ind <- LDDM_cleaning04_d2_calc1 %>%
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
filter(rt>=0.1 & rt>(mean_rt - 3*sd_rt) & rt<(mean_rt + 3*sd_rt)) %>% #remove trials less than 100ms
reframe(med_rt = median(rt)*1000) %>% # compute individual level median RT
mutate(rt_score = 5-(5*((log(med_rt)-log(250))/(log(1000)-log(250)))),
rt_score_truncated = 5-(5*((log(med_rt)-log(500))/(log(3000)-log(500)))))

LDDM_cleaning04_d2_calc3 <- LDDM_cleaning04_d2_calc1 %>%
full_join(LDDM_cleaning04_d2_ind, by = 'subj_idx') %>%
full_join(LDDM_cleaning04_d2_calc2, by='subj_idx') %>%
group_by(subj_idx) %>% # per subject
mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
reframe(flanker_score_list_d2 = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score, accuracy_score),
flanker_score_list_d2_truncated = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score_truncated, accuracy_score),
accuracy = mean(total_accuracy_perc)) %>% # if accuracy is above 80% then add accuracy and rt score
transmute(ID=subj_idx, flanker_score_list_d2=flanker_score_list_d2, flanker_score_list_d2_truncated=flanker_score_list_d2_truncated)
group_by(ID) %>% # per subject
reframe(flanker_score = mean(flanker_score_list_d2),
flanker_score_truncated = mean(flanker_score_list_d2_truncated),
accuracy = mean(accuracy))

LDDM_cleaning04_d2_accuracy_by_condition <- LDDM_cleaning04_d2_calc1 %>%
full_join(LDDM_cleaning04_d2_ind, by = 'subj_idx') %>%
full_join(LDDM_cleaning04_d2_calc2, by='subj_idx') %>%
group_by(subj_idx, stim) %>% # per subject
mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
reframe(accuracy_by_stim = mean(total_accuracy_perc)) %>%
pivot_wider(names_from=stim, values_from=accuracy_by_stim, id_cols=subj_idx, names_prefix="accuracy_")
mutate(ID=subj_idx,
accuracy_congruent_log = log(accuracy_congruent))

LDDM_cleaning04_d2_calc4 <- full_join(LDDM_cleaning04_d2_calc3, LDDM_cleaning04_d2_accuracy_by_condition)
select(-subj_idx)

```

Only 20 trials

```

# Setup the dataset with the needed variables (eg., ID, block # 1-11, trial # 1-30 per block)
LDDM_cleaning04_d2_calc1_short_incon <- LDDM_cleaning04_d2_calc1 %>%
group_by(subj_idx) %>%

```

```

filter(stim=="incongruent") %>%
slice_head(n=8)

LDDM_cleaning04_d2_calc1_short_con <- LDDM_cleaning04_d2_calc1 %>%
  group_by(subj_idx) %>%
  filter(stim=="congruent") %>%
  slice_head(n=12)

LDDM_cleaning04_d2_calc1_short <- rbind(LDDM_cleaning04_d2_calc1_short_incon, LDDM_cleaning04_d2_calc1_short_con) %>%
  arrange(subj_idx)

# Calculate the accuracy per subject (according to NIH toolbox this should be the number of correct responses)
LDDM_cleaning04_d2_calc2_short <- LDDM_cleaning04_d2_calc1_short %>%
  group_by(subj_idx) %>%
  reframe(accuracy_score = sum(response==1)*(5/length(trial)))

LDDM_cleaning04_d2_ind_short <- LDDM_cleaning04_d2_calc1_short %>%
  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
  filter(rt>=0.1 & rt<(mean_rt + 3*sd_rt)) %>% #remove trials less than 100ms
  reframe(med_rt = median(rt)*1000) %>% # compute individual level median RT
  mutate(rt_score = 5-(5*((log(med_rt)-log(250))/(log(1000)-log(250)))),
         rt_score_truncated = 5-(5*((log(med_rt)-log(500))/(log(3000)-log(500)))))

LDDM_cleaning04_d2_calc3_short <- LDDM_cleaning04_d2_calc1_short %>%
  full_join(LDDM_cleaning04_d2_ind, by = 'subj_idx') %>%
  full_join(LDDM_cleaning04_d2_calc2, by='subj_idx') %>%
  group_by(subj_idx) %>% # per subject
  mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
  reframe(flanker_score_list_d2 = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score, accuracy_score),
         flanker_score_list_d2_truncated = 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 rt score
  transmute(ID=subj_idx, flanker_score_list_d2=flanker_score_list_d2, flanker_score_list_d2_truncated=flanker_score_list_d2_truncated)
  group_by(ID) %>% # per subject
  reframe(flanker_score = mean(flanker_score_list_d2),
         flanker_score_truncated = mean(flanker_score_list_d2_truncated),
         accuracy = mean(accuracy))

```

Day 3

```

# t.test(rt ~ stim, LDDM_cleaning04_d3_calc1)

# Setup the dataset with the needed variables (eg., ID, block # 1-11, trial # 1-30 per block)
LDDM_cleaning04_d3_calc1 <- LDDM_cleaning04_fullbeh_d3 %>%
  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),rep(10,30),rep(11,30)))
  group_by(subj_idx, block) %>%
  mutate(trial = 1:30) %>%
  filter(rt<999)

```

```

# Calculate the accuracy per subject (according to NIH toolbox this should be the number of correct responses)
LDDM_cleaning04_d3_calc2 <- LDDM_cleaning04_d3_calc1 %>%
  group_by(subj_idx) %>%
  reframe(accuracy_score = sum(response==1)*(5/length(trial)))

LDDM_cleaning04_d3_ind <- LDDM_cleaning04_d3_calc1 %>%
  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
  filter(rt>=0.1 & rt>(mean_rt - 3*sd_rt) & rt<(mean_rt + 3*sd_rt)) %>% #remove trials less than 100ms
  reframe(med_rt = median(rt)*1000) %>% # compute individual level median RT
  mutate(rt_score = 5-(5*((log(med_rt)-log(250))/(log(1000)-log(250)))),
         rt_score_truncated = 5-(5*((log(med_rt)-log(500))/(log(3000)-log(500)))))

LDDM_cleaning04_d3_calc3 <- LDDM_cleaning04_d3_calc1 %>%
  full_join(LDDM_cleaning04_d3_ind, by = 'subj_idx') %>%
  full_join(LDDM_cleaning04_d3_calc2, by='subj_idx') %>%
  group_by(subj_idx) %>% # per subject
  mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
  reframe(flanker_score_list_d3 = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score, accuracy_score),
         flanker_score_list_d3_truncated = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score_truncated, accuracy_score_truncated),
         accuracy = mean(total_accuracy_perc)) %>% # if accuracy is above 80% then add accuracy and rt score
  transmute(ID=subj_idx, flanker_score_list_d3=flanker_score_list_d3, flanker_score_list_d3_truncated=flanker_score_list_d3_truncated)
  group_by(ID) %>% # per subject
  reframe(flanker_score = mean(flanker_score_list_d3),
         flanker_score_truncated = mean(flanker_score_list_d3_truncated),
         accuracy = mean(accuracy))

LDDM_cleaning04_d3_accuracy_by_condition <- LDDM_cleaning04_d3_calc1 %>%
  full_join(LDDM_cleaning04_d3_ind, by = 'subj_idx') %>%
  full_join(LDDM_cleaning04_d3_calc2, by='subj_idx') %>%
  group_by(subj_idx, stim) %>% # per subject
  mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
  reframe(accuracy_by_stim = mean(total_accuracy_perc)) %>%
  pivot_wider(names_from=stim, values_from=accuracy_by_stim, id_cols=subj_idx, names_prefix="accuracy_")
  mutate(ID=subj_idx,
         accuracy_congruent_log = log(accuracy_congruent))

LDDM_cleaning04_d3_calc4 <- full_join(LDDM_cleaning04_d3_calc3, LDDM_cleaning04_d3_accuracy_by_condition)
select(-subj_idx)

```

Only 20 trials

```

# Setup the dataset with the needed variables (eg., ID, block # 1-11, trial # 1-30 per block)
LDDM_cleaning04_d3_calc1_short_incon <- LDDM_cleaning04_d3_calc1 %>%
  group_by(subj_idx) %>%
  filter(stim=="incongruent") %>%
  slice_head(n=8)

LDDM_cleaning04_d3_calc1_short_con <- LDDM_cleaning04_d3_calc1 %>%

```



```

group_by(subj_idx) %>%
  filter(stim=="congruent") %>%
  slice_head(n=12)

LDDM_cleaning04_d3_calc1_short <- rbind(LDDM_cleaning04_d3_calc1_short_incon, LDDM_cleaning04_d3_calc1_short_con)
  arrange(subj_idx)

# Calculate the accuracy per subject (according to NIH toolbox this should be the number of correct responses)
LDDM_cleaning04_d3_calc2_short <- LDDM_cleaning04_d3_calc1_short %>%
  group_by(subj_idx) %>%
  reframe(accuracy_score = sum(response==1)*(5/length(trial)))

LDDM_cleaning04_d3_ind_short <- LDDM_cleaning04_d3_calc1_short %>%
  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
  filter(rt>=0.1 & rt>(mean_rt - 3*sd_rt) & rt<(mean_rt + 3*sd_rt)) %>% #remove trials less than 100ms
  reframe(med_rt = median(rt)*1000) %>% # compute individual level median RT
  mutate(rt_score = 5-(5*((log(med_rt)-log(250))/(log(1000)-log(250)))),
         rt_score_truncated = 5-(5*((log(med_rt)-log(500))/(log(3000)-log(500)))))

LDDM_cleaning04_d3_calc3_short <- LDDM_cleaning04_d3_calc1_short %>%
  full_join(LDDM_cleaning04_d3_ind, by = 'subj_idx') %>%
  full_join(LDDM_cleaning04_d3_calc2, by='subj_idx') %>%
  group_by(subj_idx) %>% # per subject
  mutate(total_accuracy_perc = sum(response==1)/length(response)) %>% # compute total accuracy percentage
  reframe(flanker_score_list_d3 = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score, accuracy_score),
         flanker_score_list_d3_truncated = if_else(total_accuracy_perc>0.8, accuracy_score+rt_score_truncated, accuracy_score),
         accuracy = mean(total_accuracy_perc)) %>%# if accuracy is above 80% then add accuracy and rt
  transmute(ID=subj_idx, flanker_score_list_d3=flanker_score_list_d3, flanker_score_list_d3_truncated=flanker_score_list_d3_truncated)
  group_by(ID) %>% # per subject
  reframe(flanker_score = mean(flanker_score_list_d3),
         flanker_score_truncated = mean(flanker_score_list_d3_truncated),
         accuracy = mean(accuracy))

```

Closing out

In this step, go ahead and close out of the file and quit R without saving the work space.

```
LDDM_cleaning04_d1_calc4 <- full_join(LDDM_cleaning04_d1_calc4, sttr_all_data, by="ID")
LDDM_cleaning04_d2_calc4 <- full_join(LDDM_cleaning04_d2_calc4, sttr_all_data, by="ID")
LDDM_cleaning04_d3_calc4 <- full_join(LDDM_cleaning04_d3_calc4, sttr_all_data, by="ID")

save(LDDM_cleaning04_fullbeh_d1, file=here("./work/data/LDDM_cleaning04_fullbeh_d1.RData"))
save(LDDM_cleaning04_fullbeh_d2, file=here("./work/data/LDDM_cleaning04_fullbeh_d2.RData"))
save(LDDM_cleaning04_fullbeh_d3, file=here("./work/data/LDDM_cleaning04_fullbeh_d3.RData"))

save(LDDM_cleaning04_d1_calc4, LDDM_cleaning04_d1_calc1, LDDM_cleaning04_d1_calc3_short, LDDM_cleaning04_d1_calc4)
save(LDDM_cleaning04_d2_calc4, LDDM_cleaning04_d2_calc1, LDDM_cleaning04_d2_calc3_short, LDDM_cleaning04_d2_calc4)
```



```
save(LDDM_cleaning04_d3_calc4, LDDM_cleaning04_d3_calc1, LDDM_cleaning04_d3_calc3_short, LDDM_cleaning04_d3_calc2,
     # renv::snapshot() #Take a snapshot of environment
```