Fall 2017 Data Cleaning

Amy Rae Fox

11/2/2021

The purpose of this file is processing the combined data files for Fall 2017 into study-level files that contain only valid data for analysis, excluding invalid sessions and conditions.

Data is imported from 2 files, indicating two levels of analysis: participants and blocks (item-level). **Note: mouse-cursor data contained in final_mouse_blocks.json file is not handled here.**

```
#IMPORT DATA
df_participants <- fromJSON("combined_files/final_participants.json")
df_blocks <- fromJSON('combined_files/final_blocks.json')

#add term indicator
df_participants$term <- "fall17"
df_blocks$term <- "fall17"
```

```
#create factors in PARTICIPANTS
df_participants <- df_participants %>%
  select(subject,session,term,condition,  #re-arrange columns
         ts_n, tt_n,triangular_score,
         os_n, ot_n,orthogonal_score,
         explicit,impasse,axis,
         triangular_time, totalTime, ts_t, tt_t,
         attn_check,
         native_language, year, major, country, sex, age
         ) %>% #reorder columns
  mutate( #create factors and remove extraneous ""
    subject=factor(subject),
    condition=factor(condition),
    session=factor(session),
    term=factor(term),
    explicit=factor(explicit),
    axis=factor(axis),
    impasse=factor(impasse),
    sex = as.factor(gsub('"',"",sex)),
    age = as.double(gsub('"',"",age)),
    country = gsub('"',"",country),
    major = gsub('"',"",major),
    year = gsub('"',"",year),
    native_language = gsub('"',"",native_language),
)
```

```
df_blocks <- df_blocks %>%
  select( #reorder columns
    subject, session, term, condition,
    q,question,answer,rt,
```

```
    correct, orth_correct,
    explicit, impasse, axis) %>%
  mutate(
    subject=factor(subject),
    condition=factor(condition),
    session=factor(session),
    term=factor(term),
    explicit=factor(explicit),
    axis=factor(axis),
    impasse=factor(impasse),
    q=factor(q),
    question=factor(question)
  )
```

Sessions

The (string) session code is entered by the participant based on instructions given by the experimenter, and documents the data-collection session (eg. in-person at a particular time). This code is also used by the experimenter to differentiate test or expert data collection runs.

```
#MANUALLY INSPECT sessions
df_participants %>% group_by(session) %>%
  summarize(n=n())
```

```
## # A tibble: 16 x 2
##    session      n
##    <fct>    <int>
##  1 111          1
##  2 alfa        19
##  3 alpha        1
##  4 bravo        7
##  5 charlie     13
##  6 delta        5
##  7 echo         9
##  8 foxtrot      8
##  9 golf         7
## 10 hotel       21
## 11 india       23
## 12 juliet      11
## 13 kilo        20
## 14 lima        24
## 15 mike        19
## 16 pinecone     2
```

```
#manually recode sessions in participants
df_participants$session <- recode(df_participants$session,
                                  'alpha'='alfa',
                                  '111'="XTRA")

#manually recode sessions in blocks
df_blocks$session <- recode(df_blocks$session,
                                  'alpha'='alfa',
                                  '111'="XTRA")
```

```
df_participants %>% group_by(session) %>%
  arrange(desc(session)) %>%
  summarize(n=n())
```

```
## # A tibble: 15 x 2
##    session      n
##    <fct>    <int>
##  1 XTRA         1
##  2 alfa        20
##  3 bravo        7
##  4 charlie     13
##  5 delta        5
##  6 echo         9
##  7 foxtrot      8
##  8 golf         7
##  9 hotel       21
## 10 india       23
## 11 juliet      11
## 12 kilo        20
## 13 lima        24
## 14 mike        19
## 15 pinecone     2
```

In Fall 2017, 14 data collection sessions were used, from ALFA -> PINECONE. Participants who mispelled their sessions have been manually recoded, and one participant erroneously entered their condition code as their session code, and this entry is corrected to 'XTRA'.

No data need to be excluded based on SESSION CODE.

Conditions

The three digit condition code is entered by the participant based on instructions given by the experimenter, and determines the stimulus that the participant experiences during the study.

```
df_participants %>% group_by(condition) %>%
  summarize(n=n())
```

```
## # A tibble: 8 x 2
##   condition     n
##   <fct>     <int>
## 1 111          27
## 2 112          10
## 3 113           6
## 4 121          27
## 5 211          30
## 6 221          30
## 7 311          30
## 8 321          30
```

```
#SET CONDITION FACTORS FOR EACH STUDY
#SGC3A is the simple insight study, control (111) vs impasse (121)
f_sgc3a <- c(111,121)

#SGC3B is the factorial insight study (111 control, 121 insight, 211 static, 221 static-impasse, 311 ix
```

```
f_sgc3b <- c(111,121,211,221,311,321)

#SGC4 is the gridlines study 111, 112, 113
f_sgc4 <- c(111,112,113)
```

In Fall 2017, data were gathered for three study designs: SGC3A (simple insight vs. control), SGC3B (partial data collected: full factorial insight vs. explicit) and SGC4(partial data collected:gridlines).

Finally, data from the master participants and blocks files are segregated into separate files for each individual study, separated by condition.

```
#SEPARATE PARTICIPANTS FILES
df_sgc3a <- df_participants %>% filter (condition %in% f_sgc3a)
df_sgc3a %>% group_by(condition) %>%
  summarize(n=n())
```

```
## # A tibble: 2 x 2
##   condition     n
##   <fct>     <int>
## 1 111          27
## 2 121          27
```

```
write.csv(df_sgc3a,"study_files/fall17_sgc3a_participants.csv", row.names = FALSE)

df_sgc3b <- df_participants %>% filter (condition %in% f_sgc3b)
df_sgc3b %>% group_by(condition) %>%
  summarize(n=n())
```

```
## # A tibble: 6 x 2
##   condition     n
##   <fct>     <int>
## 1 111          27
## 2 121          27
## 3 211          30
## 4 221          30
## 5 311          30
## 6 321          30
```

```
write.csv(df_sgc3b,"study_files/fall17_sgc3b_participants.csv", row.names = FALSE)

df_sgc4 <- df_participants %>% filter (condition %in% f_sgc4)
df_sgc4 %>% group_by(condition) %>%
  summarize(n=n())
```

```
## # A tibble: 3 x 2
##   condition     n
##   <fct>     <int>
## 1 111          27
## 2 112          10
## 3 113           6
```

```
write.csv(df_sgc4,"study_files/fall17_sgc4_participants.csv", row.names = FALSE)

#SEPARATE BLOCKS FILES
df_sgc3a <- df_blocks %>% filter (condition %in% f_sgc3a)
df_sgc3a %>% group_by(condition) %>%
  summarize(n=n())
```

```
## # A tibble: 2 x 2
##   condition     n
##   <fct>     <int>
## 1 111         405
## 2 121         405
```

```
write.csv(df_sgc3a,"study_files/fall17_sgc3a_blocks.csv", row.names = FALSE)

df_sgc3b <- df_blocks %>% filter (condition %in% f_sgc3b)
df_sgc3b %>% group_by(condition) %>%
  summarize(n=n())
```

```
## # A tibble: 6 x 2
##   condition     n
##   <fct>     <int>
## 1 111         405
## 2 121         405
## 3 211         450
## 4 221         450
## 5 311         450
## 6 321         450
```

```
write.csv(df_sgc3b,"study_files/fall17_sgc3b_blocks.csv", row.names = FALSE)

df_sgc4 <- df_blocks %>% filter (condition %in% f_sgc4)
df_sgc4 %>% group_by(condition) %>%
  summarize(n=n())
```

```
## # A tibble: 3 x 2
##   condition     n
##   <fct>     <int>
## 1 111         405
## 2 112         150
## 3 113          90
```

```
write.csv(df_sgc4,"study_files/fall17_sgc4_blocks.csv", row.names = FALSE)
```