Winter 2022 SGC 3A Data Cleaning

Amy Rae Fox

02/22/2022

The purpose of this file is processing the combined data files for Winter 2021 into study-level files that contain only valid data for analysis, excluding invalid sessions and participants

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/winter22_sgc3a_final_participants.json")
df_items <- fromJSON('combined_files/winter22_sgc3a_final_items.json')

#add term indicator
df_participants$term <- "winter22"
df_items$term <- "winter22"

#DEFINE SGC_3A validity crieria
sessions <- c('wi22sona') #SGC3A second online replication on SONA
conditions <-c(111,121) #2 conditions
violation_threshold = 3 #number of allowable browser violations
effort_exclusion = c("I didn't try very hard, or rushed through the questions")

#placeholder for excluding participants
ex_participants = data.frame()
```

```
#create factors in PARTICIPANTS
df_participants <- df_participants %>%
  mutate( #create factors and remove extraneous ""
    subject=factor(subject),
    condition=factor(condition),
    session=factor(session),
    term=factor(term),
    sex = as.factor(gender),
    age = as.integer(age),
    country = gsub('"',"",country),
    year = factor(schoolyear),
    native_language = factor(language)
   ) %>% select(
     -"_id"
   )
```

```
df_items <- df_items %>%
  mutate(
    subject=factor(subject),
    condition=factor(condition),
    session=factor(session),
    term=factor(term),
    explicit=factor(explicit),
    grid=factor(grid),
    impasse=factor(impasse),
    q=factor(q)
  ) %>% select(
    -"_id"
   )
```

Data Validation

Completion Status

Starting with Winter 2022, data are saved to the database even if the subject's browser did not meet minimum specifications (at which point they are prompted to change browsers, or end the study). This allows us to learn about the browsers, screen sizes and OS that (potential) subjects are using. However, these data are *not* exported from the database for analysis (see flatten.js and status.js scripts). Thus, only subjects who successfully completed the entire study are included in this file.

```
#MANUALLY INSPECT status
df_participants %>% group_by(status) %>%
  dplyr::summarize(n=n())
```

```
## # A tibble: 1 x 2
##   status      n
##   <chr>   <int>
## 1 success    82
```

```
#DISCARD participants from invalid sessions
exclude_status <- df_participants %>%
          filter(status != "success") %>%
          mutate(reason="invalid-status")

ex_participants <- rbind(ex_participants, exclude_status)
rm(exclude_status)

df_participants <- df_participants %>%
  filter( ! subject %in% ex_participants$subject)
```

No data need to be excluded on account of completion status.

Conditions

Participants are randomly assigned to an experimental condition when starting the study. Here we validate that only conditions for the current study are included in this dataset.

```
#MANUALLY INSPECT conditions
df_participants %>% group_by(condition) %>%
  dplyr::summarize(n=n())
```

```
## # A tibble: 2 x 2
##   condition     n
##   <fct>     <int>
## 1 111          38
## 2 121          44
```

Data from conditions *not* corresponding to valid conditions should be discarded.

```
#DISCARD participants from conditions invalid for this study
exclude_condition <- df_participants %>%
          filter(!condition %in% conditions) %>%
          mutate(reason="invalid-condition")

ex_participants <- rbind(ex_participants, exclude_condition)
rm(exclude_condition)

df_participants <- df_participants %>%
  filter( ! subject %in% ex_participants$subject)
```

No data need to be excluded on account of condition.

Sessions

The (string) `session code` is embedded in the URL querystring by the experimenter to differentiate testing sessions in SONA from demo and other environment setup tasks.

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

```
## # A tibble: 1 x 2
##   session      n
##   <fct>    <int>
## 1 wi22sona    82
```

Data from sessions not corresponding to valid sessions should be discarded.

```
#DISCARD participants from invalid sessions
exclude_session <- df_participants %>%
          filter(!session %in% sessions) %>%
          mutate(reason="invalid-session")

ex_participants <- rbind(ex_participants, exclude_session)
rm(exclude_session)

df_participants <- df_participants %>%
  filter( ! subject %in% ex_participants$subject)
```

No data need to be excluded on account of session.

Browser Interaction Violations

Browser interaction data is recorded by jspsych allowing us to determine if subjects violate our instructions not to leave the browser tab (or exit fullscreen mode) during test. These incidents are recorded in jspsych interaction data object, and the number of violations is counted and added to the participant data file.

Due to eccentricity of the browser events captured, 1-2 browser violations can be captured even if the subject did not leave the browser window (eg. in case of resizing window to meet minimum requirements.)

```
#MANUALLY INSPECT violations
df_participants %>% group_by(violations) %>%
  dplyr::summarize(n=n())
```

```
## # A tibble: 6 x 2
##   violations     n
##        <dbl> <int>
## 1        1      55
## 2        1.5     3
## 3        2      15
## 4        2.5     1
## 5        3       6
## 6        3.5     2
```

```
#DISCARD participants exceeding the threshold of browser interaction violations
exclude_violations <- df_participants %>%
          filter(violations > violation_threshold) %>%
          mutate(reason="exceeded-violations")

ex_participants <- rbind(ex_participants, exclude_violations)
rm(exclude_violations)

df_participants <- df_participants %>%
  filter( ! subject %in% ex_participants$subject)
```

Two participants were excluded for exceeding the maximum allowed number of browser interaction violations.

Effort

To assist in mitigating increased noise in data collected asynchronously from the UCSD student subject pool, we added explicit ratings of how much effort the participant expended on the task. This question was implemented as a multiple-choice drop-down on an ‘Effort’ page prior to the ‘Demographics’ survey at the end of the study. Subjects were given four options : (1) I tried my best on each question, (2) I tried my best on most questions, (3) I started out trying hard, but gave up at some point, (4) I didn’t try very hard, or rushed through the questions.

```
#MANUALLY INSPECT effort
df_participants %>% group_by(effort) %>%
  dplyr::summarize(n=n())
```

```
## # A tibble: 4 x 2
##   effort                                                         n
##   <chr>                                                      <int>
## 1 I didn't try very hard, or rushed through the questions     3
## 2 I started out trying hard, but gave up at some point        6
## 3 I tried my best on each question                           50
## 4 I tried my best on most questions                          21
```

Participants answering with options *I didn't try very hard, or rushed through the questions* are excluded for all analyses. Participants answering *I started out trying hard, but gave up at some point* **are** included in analysis, as we are interested in problem sovling strategy under the condition of losing motivation.

```
#DISCARD participants who indicated they did not expend adequate effort on the study
exclude_effort <- df_participants %>%
          filter(effort %in% effort_exclusion) %>%
          mutate(reason="selfrated-effort")

ex_participants <- rbind(ex_participants, exclude_effort)
rm(exclude_effort)

df_participants <- df_participants %>%
  filter( ! subject %in% ex_participants$subject)
```

Three participants are excluded for low (self-rated) effort.

Attention Check

The 6th question in the study is non-discriminatory (can easily get correct answer regardless of strategy) and serves as an attention check question.

```
#MANUALLY INSPECT attention
df_participants %>% group_by(attn_check) %>%
  dplyr::summarize(n=n())
```

```
## # A tibble: 2 x 2
##   attn_check     n
##   <lgl>      <int>
## 1 FALSE          9
## 2 TRUE          68
```

Participants who answered the attention check question incorrectly should be excluded.

```
#DISCARD participants who indicated they did not expend adequate effort on the study
exclude_attn <- df_participants %>%
          filter(attn_check == FALSE) %>%
          mutate(reason="failed-attnchk")

ex_participants <- rbind(ex_participants, exclude_attn)
rm(exclude_attn)

df_participants <- df_participants %>%
  filter( ! subject %in% ex_participants$subject)
```

Nine participants are excluded for failing the attention check question.

Items

Finally, we need to discard item_level data for excluded participants.

```
ex_items <- df_items %>%
  filter (subject %in% ex_participants$subject)

df_items <- df_items %>%
  filter (!subject %in% ex_participants$subject )
```

Data Export

Save Exclusions

For transparency, we save and identify the excluded data.

```
write.csv(ex_participants,"study_files/excluded_participants.csv", row.names = FALSE)
write.csv(ex_items,"study_files/excluded_items.csv", row.names = FALSE)
```

Analysis-Ready Files

```
#save participant file

write.csv(df_participants,"study_files/winter22_sgc3a_participants.csv", row.names = FALSE)

#save item file
write.csv(df_items,"study_files/winter22_sgc3a_items.csv", row.names = FALSE)
```