# Winter 2022 SGC 4B Data Cleaning

Amy Rae Fox

04/07/2022

## Contents



*The purpose of this file is processing the combined data files for Winter 2022 into 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("input/winter22_sgc4b_final_participants.json")
df_items <- fromJSON('input/winter22_sgc4b_final_items.json')

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

#DEFINE SGC_4A validity crieria
sessions <- c('wi22sona') #SGC4B second online replication on SONA
conditions <-c(11111,1112,1113) #3 conditions
violation_threshold = 3 #number of allowable browser violations
effort_exclusion = c("I didn't try very hard, or rushed through the questions", "I started out trying hard
n_items = 15 #fifteen items is complete dataset per participant

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

```
#create factors in PARTICIPANTS
df_participants <- df_participants %>%
  mutate( #create factors and remove extraneous ""
    subject=as.character(subject),
    condition=as.character(condition),
    study = factor(study),
    session = factor(session),
    exp_id = factor(exp_id),
```

```r
    sona_id = as.character(sona_id),
    pool = factor(pool),
    mode = factor(mode),
    attn_check = factor(attn_check),
    status=factor(status),
    term=factor(term),
    gender = as.factor(gender),
    age = as.integer(age),
    country = gsub('"',"",country),
    year = factor(schoolyear),
    major = factor(major),
    browser = factor(browser),
    os = factor(os),
    native_language = factor(language),
    totaltime_m = totaltime/1000/60,
   ) %>% select( #order cols
    subject,
    study,
    condition,
    session,
    exp_id,
    sona_id,
    pool,
    mode,
    attn_check,
    # explanation,
    effort,
    difficulty,
    confidence,
    enjoyment,
    other,
    age,
    country,
    language,
    schoolyear,
    major,
    gender,
    disability,
    browser,
    width,
    height,
    os,
    starttime,
    status,
    term,
    violations,
    absolute_score,
    discriminant_score,
    tri_score,
    orth_score,
    other_score,
    blank_score,
    totaltime_m
   )

#ADD CONTROL CONDITION MOVED FROM SGC4A
```

```r
control_participants<- read.csv("input/winter22_sgc4b_CONTROL_participants.csv") %>% select(-explanation)
  condition= as.character(condition),
  subject = as.character(subject))
df_participants <- rbind(df_participants, control_participants) %>% mutate(
  sona_id = factor(sona_id),
  subject=factor(subject),
  condition=factor(condition),
)

#remove temps
rm(control_participants)
```

```r
df_items <- df_items %>%
  mutate(
    # subject=factor(subject),
    # condition=factor(condition),
    pool=factor(pool),
    mode = factor(mode),
    # explicit=factor(explicit),
    # impasse = factor(impasse),
    # grid = factor(grid),
    # mark = factor(mark),
    # ixn = factor(ixn),
    term=factor(term),
    relation = factor(relation),
    block = factor(block),
    correct = factor(correct),
    q=factor(q),
    rt_s = rt/1000,
    time_elapsed_m = time_elapsed/1000/60
  ) %>% select(
     subject,
     study,
     term,
     pool,
     mode,
     block,
     explicit,
     impasse,
     grid,
     mark,
     ixn,
     gwidth,
     gheight,
     graph,
     time_elapsed_m,
     question,
     relation,
     q,
     correct,
     discriminant,
     tri_score,
     orth_score,
     other_score,
     blank_score,
     answer,
```

```
     rt_s,
     condition
   )

#ADD CONTROL CONDITION MOVED FROM SGC4A
control_items<- read.csv("input/winter22_sgc4b_CONTROL_items.csv")
df_items <- rbind(df_items, control_items) %>% mutate(
    subject=factor(subject),
    condition=factor(condition),
    explicit=factor(explicit),
    impasse = factor(impasse),
    grid = factor(grid),
    mark = factor(mark),
    ixn = factor(ixn),
)

#remove temps
rm(control_items)
```

# Data Validation

## Exclusions

### 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
##   <fct>   <int>
## 1 success   368
```

368 successfully completed the study.

```
#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: 3 x 2
##   condition     n
##   <fct>     <int>
## 1 11111       119
## 2 1112        135
## 3 1113        114
```

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   368
```

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: 17 x 2
##    violations     n
##         <dbl> <int>
##  1        1     248
##  2        1.5    17
##  3        2      40
##  4        2.5     7
##  5        3      19
##  6        3.5     4
##  7        4      13
##  8        4.5     2
##  9        5       5
## 10        6       4
## 11        6.5     2
## 12        7       1
## 13        8       2
## 14       10       1
## 15       10.5     1
## 16       13       1
## 17       25.5     1
```

```
#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)
```

*Thirty seven 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     7
## 2 I started out trying hard, but gave up at some point       29
## 3 I tried my best on each question                          197
## 4 I tried my best on most questions                          98
```

Participants answering with options *I didn't try very hard, or rushed through the questions* or *I started out trying hard, but gave up at some point* are excluded from analysis.

```
#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)
```

*Thirty-six 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
##   <fct>      <int>
## 1 FALSE         23
## 2 TRUE         272
```

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)
```

*Twenty three participants are excluded for failing the attention check question.*

### Items

Next, 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 )
```

## Validation

After all exclusions, we are left with the following number of participants per condition:

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

```
## # A tibble: 3 x 2
##   condition     n
##   <fct>     <int>
## 1 11111        91
## 2 1112         98
## 3 1113         83
```

Finally, we need to validate we have a complete set of items for all valid participants.

```
count(df_items)[[1]] == count(df_participants)[[1]]* n_items
```

```
## [1] TRUE
```

# Participants Codebook

```
#see https://cran.r-project.org/web/packages/codebook/vignettes/codebook_tutorial.html

#ADD VARIABLE METADATA
dict <- rio::import("input/dictionary_sgc4b_participants.csv", "csv") #import data dictionary
var_label(df_participants) <- dict %>% select(VARIABLE, DESCRIPTION) %>% dict_to_list() #add variable labe

#ADD DATASET METATDATA
metadata(df_participants)$name <- "Experimental PARTICIPANTS for study SGC4B"
metadata(df_participants)$description <- "Data for study SGC4B summarized at PARTICIPANT  level"
metadata(df_participants)$creator <- "Amy Rae Fox"
metadata(df_participants)$contact <- "amyraefox@gmail.com"
```

```
#{r, eval = checkMode() == "pdf"} #ONLY FOR PDF KNIT
codebook::skim_codebook(df_participants)
```

Table 1: Data summary

| | |
|---|---|
| Name | data |
| Number of rows | 272 |
| Number of columns | 36 |

Column type frequency:

Table 1: Data summary

| | |
|---|---|
| character | 7 |
| factor | 15 |
| numeric | 14 |
| Group variables | None |

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| effort | 0 | 1 | 32 | 33 | 0 | 2 | 0 |
| other | 0 | 1 | 0 | 414 | 166 | 96 | 0 |
| country | 0 | 1 | 2 | 24 | 0 | 36 | 0 |
| language | 0 | 1 | 6 | 9 | 0 | 8 | 0 |
| schoolyear | 0 | 1 | 5 | 6 | 0 | 5 | 0 |
| disability | 0 | 1 | 0 | 72 | 115 | 27 | 0 |
| starttime | 0 | 1 | 24 | 24 | 0 | 272 | 0 |

**Variable type: factor**

| skim_variable | n_missing | complete_rate | ordered | n_unique | top_counts |
|---|---|---|---|---|---|
| subject | 0 | 1 | FALSE | 272 | 03D: 1, 04T: 1, 0EJ: 1, 0FH: 1 |
| study | 0 | 1 | FALSE | 1 | SGC: 272 |
| condition | 0 | 1 | FALSE | 3 | 111: 98, 111: 91, 111: 83 |
| session | 0 | 1 | FALSE | 1 | wi2: 272 |
| exp_id | 0 | 1 | FALSE | 2 | 221: 157, 221: 115 |
| sona_id | 0 | 1 | FALSE | 259 | 422: 3, 325: 2, 354: 2, 362: 2 |
| pool | 0 | 1 | FALSE | 1 | son: 272 |
| mode | 0 | 1 | FALSE | 1 | asy: 272 |
| attn_check | 0 | 1 | FALSE | 1 | TRU: 272, FAL: 0 |
| major | 0 | 1 | FALSE | 7 | Soc: 178, Bio: 39, Mat: 17, Hum: 14 |
| gender | 0 | 1 | FALSE | 3 | Fem: 175, Mal: 92, Oth: 5 |
| browser | 0 | 1 | FALSE | 1 | chr: 272 |
| os | 0 | 1 | FALSE | 4 | Mac: 173, Win: 91, Chr: 4, Win: 4 |
| status | 0 | 1 | FALSE | 1 | suc: 272 |
| term | 0 | 1 | FALSE | 1 | win: 272 |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | min | median | max | hist |
|---|---|---|---|---|---|---|---|---|
| difficulty | 0 | 1 | 3.09 | 0.99 | 1.00 | 3.00 | 5.00 | |
| confidence | 0 | 1 | 3.32 | 1.02 | 1.00 | 3.00 | 5.00 | |
| enjoyment | 0 | 1 | 3.29 | 1.17 | 1.00 | 3.00 | 5.00 | |
| age | 0 | 1 | 20.30 | 1.67 | 12.00 | 20.00 | 28.00 | |
| width | 0 | 1 | 1536.99 | 250.13 | 1184.00 | 1440.00 | 2560.00 | |
| height | 0 | 1 | 811.59 | 119.28 | 644.00 | 789.00 | 1361.00 | |
| violations | 0 | 1 | 1.32 | 0.60 | 1.00 | 1.00 | 3.00 | |
| absolute_score | 0 | 1 | 2.36 | 3.89 | 0.00 | 0.00 | 12.00 | |
| discriminant_score | 0 | 1 | -5.23 | 7.77 | -12.33 | -8.38 | 12.00 | |
| tri_score | 0 | 1 | 3.32 | 5.04 | 0.00 | 1.00 | 15.00 | |

| skim_variable | n_missing | complete_rate | mean | sd | min | median | max | hist |
|---|---|---|---|---|---|---|---|---|
| orth_score | 0 | 1 | 9.67 | 5.08 | 0.00 | 11.00 | 15.00 | |
| other_score | 0 | 1 | 2.65 | 3.02 | 0.00 | 2.00 | 14.00 | |
| blank_score | 0 | 1 | 0.27 | 0.61 | 0.00 | 0.00 | 3.00 | |
| totaltime_m | 0 | 1 | 11.10 | 5.35 | 2.49 | 9.90 | 36.42 | |

```
codebook(df_participants, #ONLY FOR HTML KNIT
         metadata_table = TRUE,
         detailed_variables = FALSE,
         detailed_scales = FALSE,
         metadata_json = FALSE,
         survey_overview = FALSE,
         missingness_report = FALSE)
```

## Items Codebook

```
#see https://cran.r-project.org/web/packages/codebook/vignettes/codebook_tutorial.html

#ADD VARIABLE METADATA
dict <- rio::import("input/dictionary_sgc4b_items.csv", "csv") #import data dictionary

var_label(df_items) <- dict %>% select(VARIABLE, DESCRIPTION) %>% dict_to_list() #add variable labels

#ADD DATASET METATDATA
metadata(df_items)$name <- "Experimental ITEMS for study SGC4B"
metadata(df_items)$description <- "Data for study SGC4B summarized at participant-item level"
metadata(df_items)$creator <- "Amy Rae Fox"
metadata(df_items)$contact <- "amyraefox@gmail.com"
```

```
#{r, eval = checkMode() == "pdf"} #ONLY FOR PDF EXPORT
skim_codebook(df_items)
```

```
## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".
```

Table 5: Data summary

| | |
|---|---|
| Name | data |
| Number of rows | 4080 |
| Number of columns | 27 |
| Column type frequency: | |
| character | 4 |
| factor | 14 |
| numeric | 9 |
| Group variables | None |

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| study | 0 | 1 | 5 | 5 | 0 | 1 | 0 |
| graph | 0 | 1 | 10 | 10 | 0 | 1 | 0 |
| question | 0 | 1 | 26 | 87 | 0 | 15 | 0 |
| answer | 0 | 1 | 0 | 21 | 97 | 147 | 0 |

**Variable type: factor**

| skim_variable | n_missing | complete_rate | ordered | n_unique | top_counts |
|---|---|---|---|---|---|
| subject | 0 | 1 | FALSE | 272 | 03D: 15, 04T: 15, 0EJ: 15, 0FH: 15 |
| term | 0 | 1 | FALSE | 1 | win: 4080 |
| pool | 0 | 1 | FALSE | 1 | son: 4080 |
| mode | 0 | 1 | FALSE | 1 | asy: 4080 |
| block | 0 | 1 | FALSE | 3 | ite: 2359, ite: 905, ite: 816 |
| explicit | 0 | 1 | FALSE | 1 | 1: 4080 |
| impasse | 0 | 1 | FALSE | 1 | 1: 4080 |
| grid | 0 | 1 | FALSE | 1 | 1: 4080 |
| mark | 0 | 1 | FALSE | 3 | 2: 1470, 1: 1365, 3: 1245 |
| ixn | 0 | 1 | FALSE | 2 | emp: 2715, 1: 1365 |
| relation | 0 | 1 | FALSE | 10 | end: 544, mee: 544, mid: 544, sta: 544 |
| q | 0 | 1 | FALSE | 15 | 1: 272, 2: 272, 3: 272, 4: 272 |
| correct | 0 | 1 | FALSE | 2 | FAL: 2898, TRU: 1182 |
| condition | 0 | 1 | FALSE | 3 | 111: 1470, 111: 1365, 111: 1245 |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | min | median | max | hist |
|---|---|---|---|---|---|---|---|---|
| gwidth | 0 | 1 | 600.00 | 0.00 | 600.00 | 600.00 | 600.00 | |
| gheight | 0 | 1 | 600.00 | 0.00 | 600.00 | 600.00 | 600.00 | |
| time_elapsed_m | 0 | 1 | 5.93 | 4.43 | 0.29 | 4.99 | 31.06 | |
| discriminant | 0 | 1 | -0.36 | 0.74 | -1.67 | -0.58 | 1.00 | |
| tri_score | 0 | 1 | 0.47 | 0.70 | 0.00 | 0.00 | 2.00 | |
| orth_score | 0 | 1 | 0.92 | 0.73 | 0.00 | 1.00 | 2.00 | |
| other_score | 0 | 1 | 0.21 | 0.65 | 0.00 | 0.00 | 9.00 | |
| blank_score | 0 | 1 | 0.02 | 0.15 | 0.00 | 0.00 | 1.00 | |
| rt_s | 0 | 1 | 31.08 | 34.05 | 1.28 | 20.40 | 509.85 | |

```
codebook(df_items,#ONLY FOR HTML EXPORT
         metadata_table = TRUE,
         detailed_variables = FALSE,
         detailed_scales = FALSE,
         metadata_json = FALSE,
         survey_overview = FALSE,
         missingness_report = FALSE)
```

# Data Export

## Save Exclusions

For transparency, we save and identify the excluded data.

```
write.csv(ex_participants,"output/excluded_participants_winter22_sgc4b.csv", row.names = FALSE)
write.csv(ex_items,"output/excluded_items_winter22_sgc4b.csv", row.names = FALSE)
```

## Analysis-Ready Files

```
#save participant file

write.csv(df_participants,"output/winter22_sgc4b_participants.csv", row.names = FALSE)

#save item file
write.csv(df_items,"output/winter22_sgc4b_items.csv", row.names = FALSE)
```