

# Fall 2021 Data Cleaning

Amy Rae Fox

1/26/2022

## Contents

<b>Summary</b>	<b>2</b>
<b>Data Validation</b>	<b>3</b>
Exclusions . . . . .	3
Validation . . . . .	6
<b>Data Export</b>	<b>7</b>

## Summary

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

- 230 subjects were recorded to study database
- 40 subjects were excluded during wrangling for failing the attention check (17%)
- 190 subjects were left for further cleaning (imported, below)
- 16 subjects were excluded for having mistakenly completed the study twice
- 1 pilot subject is excluded
- 3 subjects were excluded for invalid condition codes
- yielding 170 participants for analysis (75% of recruitment)

```
#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 ixn 3)
```

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

```
#SGC4 is the gridlines study 111, 112, 113
```

```
f_sgc4 <- c(111,112,113)
```

```
#valid condition codes
```

```
conditions <- c(111,121,211,221,311,321,112,113)
```

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 <- "fall21"
```

```
df_blocks$term <- "fall21"
```

```
#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)
  )

```

## Data Validation

### Exclusions

#### 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.

In Fall 2021, participants were instructed to enter their PID as the session field.

```

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

```

```

## # A tibble: 185 x 2
##   session          n
##   <fct>         <int>
## 1 "15862635"         1
## 2 "15994246"         1
## 3 "16114839"         1
## 4 "16132934"         1
## 5 "17012262\na17012262" 1
## 6 "a09436222"         1
## 7 "a13190800"         1
## 8 "a14821119"         1
## 9 "a14821119\na14821119" 1
## 10 "a15049392"         1
## # ... with 175 more rows

```

```
#manually recode sessions in participants
```

```
df_participants$session <- recode(df_participants$session,
  "17012262\na17012262"="17012262",
  "a14821119\na14821119"="a14821119",
  "a15049392\na15049392"="a15049392",
  "a15418907\na15418907"="a15418907",
  "a15515318\na15515318"="a15515318",
  "a15558540\na15558540"="a15558540",
  "a15897677\na15897677"="a15897677",
  "a15902241\na15902241"="a15902241",
  "a16137081\na16137081"="a16137081",
  "a16324253\na16324253"="a16324253",
  "a16328170\na16328170"="a16328170",
  "a16675361\na16675361"="a16675361",
  "a16788617\na16788617"="a16788617",
  "a16885269\na16885269"="a16885269",
  "a17082219\na17082219"="a17082219",
  "a17091192\na17091192"="a17091192",
  "a17213518\na17213518"="a17213518",
  "a16686690\n16686690\n16686690"="a16686690",
  "a15826500\na15826500\na15826500"="a15826500"
)
```

```
#manually recode sessions in blocks
```

```
df_blocks$session <- recode(df_blocks$session,
  "17012262\na17012262"="17012262",
  "a14821119\na14821119"="a14821119",
  "a15049392\na15049392"="a15049392",
  "a15418907\na15418907"="a15418907",
  "a15515318\na15515318"="a15515318",
  "a15558540\na15558540"="a15558540",
  "a15897677\na15897677"="a15897677",
  "a15902241\na15902241"="a15902241",
  "a16137081\na16137081"="a16137081",
  "a16324253\na16324253"="a16324253",
  "a16328170\na16328170"="a16328170",
  "a16675361\na16675361"="a16675361",
  "a16788617\na16788617"="a16788617",
  "a16885269\na16885269"="a16885269",
  "a17082219\na17082219"="a17082219",
  "a17091192\na17091192"="a17091192",
  "a17213518\na17213518"="a17213518",
  "a16686690\n16686690\n16686690"="a16686690",
  "a15826500\na15826500\na15826500"="a15826500"
)
```

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

```
## # A tibble: 182 x 2
##   session      n
##   <fct>      <int>
## 1 15862635      1
## 2 15994246      1
```

```
## 3 16114839      1
## 4 16132934      1
## 5 17012262      1
## 6 a09436222     1
## 7 a13190800     1
## 8 a14821119     2
## 9 a15049392     2
## 10 a15131176    1
## # ... with 172 more rows
```

Participants who have more than one entry for the PID may have participated *twice*, once via SONA and once via alternate recruitment in COGS 102A. These entries need to be removed.

## Duplicate Participants

A number of participants mistakenly completed the study twice, unsure that their SONA credit had been granted. The second (later submission) of each should be excluded.

```
#identify duplicate participants
duplicates <- df_participants %>% filter(duplicated(session)) %>% select(session)
df_duplicate_participants <- df_participants %>% filter(session %in% duplicates$session)
df_duplicate_blocks <- df_blocks %>% filter(session %in% duplicates$session)

#remove from main dataframes
df_participants <- df_participants %>% filter(!session %in% duplicates$session)
df_blocks <- df_blocks %>% filter(!session %in% duplicates$session)
```

The data from these 8 participants (16 subject records) are excluded.

## Pilot Participants

Next, one test participant (session == 'hollanlab') must be manually removed.

```
#manually remove hollan lab test participant
df_participants <- df_participants %>% filter(session != "hollanlab")
df_blocks <- df_blocks %>% filter(session != "hollanlab")

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

```
## # A tibble: 173 x 2
##   session      n
##   <fct>    <int>
## 1 15862635     1
## 2 15994246     1
## 3 16114839     1
## 4 16132934     1
## 5 17012262     1
## 6 a09436222     1
## 7 a13190800     1
## 8 a15131176     1
## 9 a15274291     1
## 10 a15378348     1
## # ... with 163 more rows
```

## 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) %>%  
  dplyr::summarize(n=n())
```

```
## # A tibble: 8 x 2  
##   condition      n  
##   <fct>      <int>  
## 1 "111"        68  
## 2 "121"        71  
## 3 "121\n121"      1  
## 4 "211"         5  
## 5 "221"        12  
## 6 "221\n221"       2  
## 7 "311"         3  
## 8 "321"        11
```

In FALL 2021, data were gathered for two studies: SGC3A (online replication), SGC3B (online replication).

A few students mistyped their condition codes. These participants should be excluded.

```
#filter out invalid condition codes  
df_participants <-df_participants %>% filter (condition %in% conditions)
```

```
df_participants %>% group_by(condition) %>%  
  arrange(desc(condition)) %>%  
  dplyr::summarize(n=n())
```

```
## # A tibble: 6 x 2  
##   condition      n  
##   <fct>      <int>  
## 1 111        68  
## 2 121        71  
## 3 211         5  
## 4 221        12  
## 5 311         3  
## 6 321        11
```

## Validation

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

### SGC\_3A

```
df_sgc3a <- df_participants %>% filter (condition %in% f_sgc3a)  
df_sgc3a %>% group_by(condition) %>%  
  dplyr::summarize(n=n())
```

```
## # A tibble: 2 x 2  
##   condition      n
```

```
##   <fct>      <int>
## 1 111         68
## 2 121         71
df_sgc3a_blocks <- df_blocks %>% filter (condition %in% f_sgc3a)

#number of items = number of subjects * 16
nrow(df_sgc3a) * 16 == nrow(df_sgc3a_blocks)

## [1] TRUE
#number of items per subject == 16 (15 items + free response)
(df_sgc3a_blocks %>% group_by(subject) %>% summarize(n = n()) %>% filter(n != 16) %>% nrow() ) == (0 )

## [1] TRUE
```

## SGC\_3B

Data collected for the factorial SGC\_3B are incomplete (ran out of time before end of SONA collection period), and considered a pilot.

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

## # A tibble: 6 x 2
##   condition      n
##   <fct>      <int>
## 1 111         68
## 2 121         71
## 3 211          5
## 4 221         12
## 5 311          3
## 6 321         11
df_sgc3b_blocks <- df_blocks %>% filter (condition %in% f_sgc3b)

#number of items = number of subjects * 16
nrow(df_sgc3b) * 16 == nrow(df_sgc3b_blocks)

## [1] TRUE
#number of items per subject == 16 (15 items + free response)
(df_sgc3b_blocks %>% group_by(subject) %>% summarize(n = n()) %>% filter(n != 16) %>% nrow() ) == (0 )

## [1] TRUE
```

## Data Export

```
#SEPARATE PARTICIPANTS FILES
write.csv(df_sgc3a,"study_files/fall21_sgc3a_participants.csv", row.names = FALSE)
write.csv(df_sgc3b,"study_files/fall21_sgc3b_participants.csv", row.names = FALSE)

#SEPARATE BLOCKS FILES
write.csv(df_sgc3a_blocks,"study_files/fall21_sgc3a_blocks.csv", row.names = FALSE)
write.csv(df_sgc3b_blocks,"study_files/fall21_sgc3b_blocks.csv", row.names = FALSE)
```