Winter 2022 SGC 5A Data Cleaning

Amy Rae Fox

04/07/2022

Contents

TODO INVESTIATE WHY ATTN CHECK IS FALSE? ALL TRI ARE 0?

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_sgc5a_final_participants.json")
df_items <- fromJSON('input/winter22_sgc5a_final_items.json')

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

#DEFINE SGC_5A validity crieria
sessions <- c('wi22sona') #SGC5A second online replication on SONA
conditions <-c(11115) #2 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=factor(subject),
    condition=factor(condition),
    pretty_condition = recode_factor(condition, "11115" = "point-click"),
    study = factor(study),
    condition = factor(condition),
    session = factor(session),
    exp_id = factor(exp_id),
    sona_id = factor(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,
    pretty_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
   )  #drop scores because they're recalculated in analysis
```

```
df_items <- df_items %>%
  mutate(
    subject=factor(subject),
    condition=factor(condition),
    pretty_condition = recode_factor(condition, "111" = "control", "121" =  "impasse"),
    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,
     condition,
     pretty_condition,
     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
   )
```

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   137
```

137 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: 1 x 2
##   condition     n
##   <fct>     <int>
## 1 11115       137
```

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   137
```

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: 10 x 2
##    violations     n
##         <dbl> <int>
##  1        1      91
##  2        1.5     5
##  3        2      19
## 4          2.5      2
## 5          3        9
## 6          3.5      3
## 7          4        3
## 8          4.5      1
## 9          5        3
## 10         6        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)
```

Eleven 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     4
## 2 I started out trying hard, but gave up at some point       7
## 3 I tried my best on each question                          86
## 4 I tried my best on most questions                         29
```

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)
```

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

TODO: WHI IS ATTENTION CHECK FALSE… CHECK THIS ### 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: 1 x 2
##   attn_check     n
##   <fct>      <int>
## 1 FALSE        115
```

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

NOTE THIS BLOCK IS NOT RUN UNTIL TROUBLESHOOT ATTN CHECK

```
#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

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: 1 x 2
##   condition     n
##   <fct>     <int>
## 1 11115       115
```

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

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

#ADD VARIABLE METADATA
dict <- rio::import("input/dictionary_sgc5a_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)$name <- "Experimental PARTICIPANTS for study SGC5A"
metadata(df_participants)$description <- "Data for study SGC5A summarized at PARTICIPANT  level"
metadata(df_participants)$creator <- "Amy Rae Fox"
metadata(df_participants)$contact <- "amyraefox@gmail.com"
```

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

Table 1: Data summary

Name	data
Number of rows	115
Number of columns	33
Column type frequency:	
character	8
factor	16
numeric	9
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
explanation	0	1	0	259	3	112	0
effort	0	1	32	33	0	2	0
other	0	1	0	403	67	42	0
country	0	1	2	24	0	21	0
language	0	1	6	9	0	7	0
schoolyear	0	1	5	6	0	5	0
disability	0	1	0	5	54	17	0
starttime	0	1	24	24	0	115	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
subject	0	1	FALSE	115	0A9: 1, 0AX: 1, 0S7: 1, 11N: 1
study	0	1	FALSE	1	SGC: 115
condition	0	1	FALSE	1	111: 115
pretty_condition	0	1	FALSE	1	poi: 115
session	0	1	FALSE	1	wi2: 115
exp_id	0	1	FALSE	1	221: 115
sona_id	0	1	FALSE	115	271: 1, 273: 1, 281: 1, 282: 1

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
pool	0	1	FALSE	1	son: 115
mode	0	1	FALSE	1	asy: 115
attn_check	0	1	FALSE	1	FAL: 115
major	0	1	FALSE	6	Soc: 68, Bio: 24, Nat: 9, Hum: 7
gender	0	1	FALSE	3	Fem: 70, Mal: 42, Oth: 3
browser	0	1	FALSE	1	chr: 115
os	0	1	FALSE	2	Mac: 71, Win: 44
status	0	1	FALSE	1	suc: 115
term	0	1	FALSE	1	win: 115

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	min	median	max	hist
difficulty	0	1	3.18	0.94	1.00	3.00	5.00	▁▅▇▇▁
confidence	0	1	3.25	0.92	1.00	3.00	5.00	▁▃▇▆▂
enjoyment	0	1	3.35	1.18	1.00	3.00	5.00	▂▅▇▇▅
age	0	1	20.67	2.44	18.00	20.00	36.00	▇▁▁▁▁
width	0	1	1559.05	285.03	1140.00	1440.00	2560.00	▂▇▂▁▁
height	0	1	828.07	134.80	687.00	792.00	1329.00	▇▅▁▁▁
violations	0	1	1.33	0.61	1.00	1.00	3.00	▇▁▂▁▁
absolute_score	0	1	0.00	0.00	0.00	0.00	0.00	▁▁▇▁▁
totaltime_m	0	1	11.30	5.60	3.15	10.08	39.03	▇▆▁▁▁

```
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_sgc5a_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)$name <- "Experimental ITEMS for study SGC5A"
metadata(df_items)$description <- "Data for study SGC5A 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)
```

Table 5: Data summary

Name	data
Number of rows	1725
Number of columns	23
Column type frequency:	
character	4
factor	15
numeric	4
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	32	57	89	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
subject	0	1	FALSE	115	0A9: 15, 0AX: 15, 0S7: 15, 11N: 15
term	0	1	FALSE	1	win: 1725
pool	0	1	FALSE	1	son: 1725
mode	0	1	FALSE	1	asy: 1725
condition	0	1	FALSE	1	111: 1725
pretty_condition	0	1	FALSE	1	111: 1725
block	0	1	FALSE	2	ite: 1380, ite: 345
explicit	0	1	FALSE	1	1: 1725
impasse	0	1	FALSE	1	1: 1725
grid	0	1	FALSE	1	1: 1725
mark	0	1	FALSE	1	1: 1725
ixn	0	1	FALSE	1	5: 1725
relation	0	1	FALSE	10	end: 230, mee: 230, mid: 230, sta: 230
q	0	1	FALSE	15	1: 115, 2: 115, 3: 115, 4: 115
correct	0	1	FALSE	1	FAL: 1725

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	5.73	4.62	0.30	4.71	36.99	▇▂▁▁▁
rt_s	0	1	30.45	36.25	0.13	19.72	646.39	▇▁▁▁▁

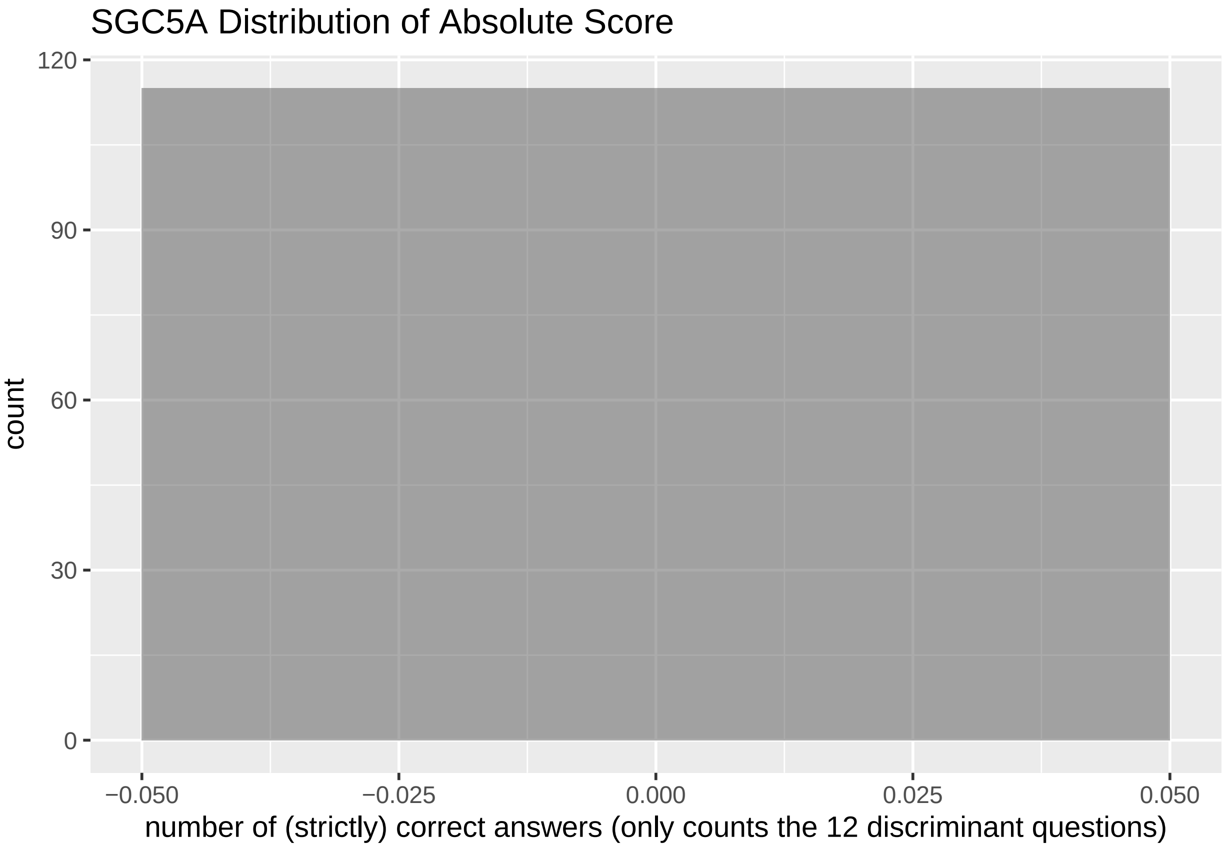
```
codebook(df_items,#ONLY FOR HTML EXPORT
         metadata_table = TRUE,
         detailed_variables = FALSE,
         detailed_scales = FALSE,
         metadata_json = FALSE,
         survey_overview = FALSE,
         missingness_report = FALSE)
```

Explore

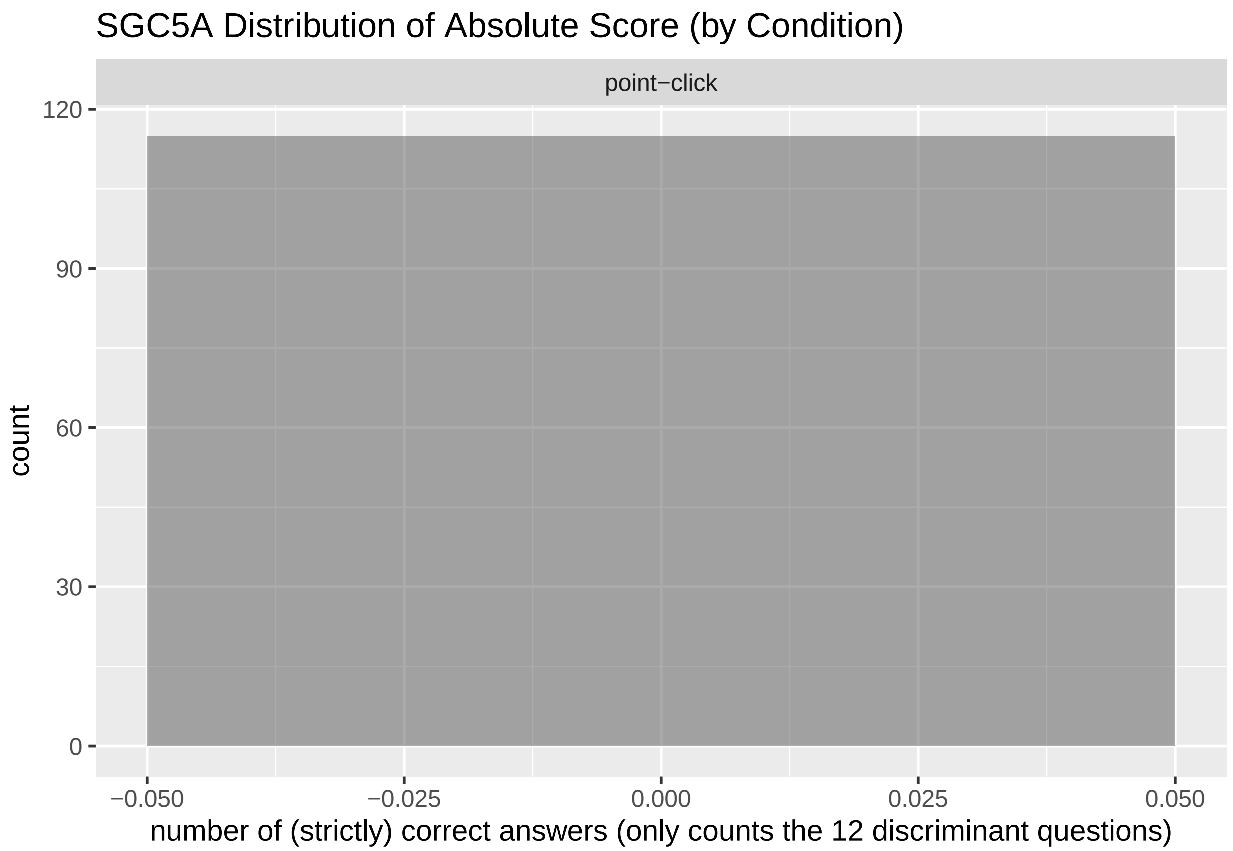
Exploration of the distribution of key response variables for validation purposes:

TODO why is absolute score 0?

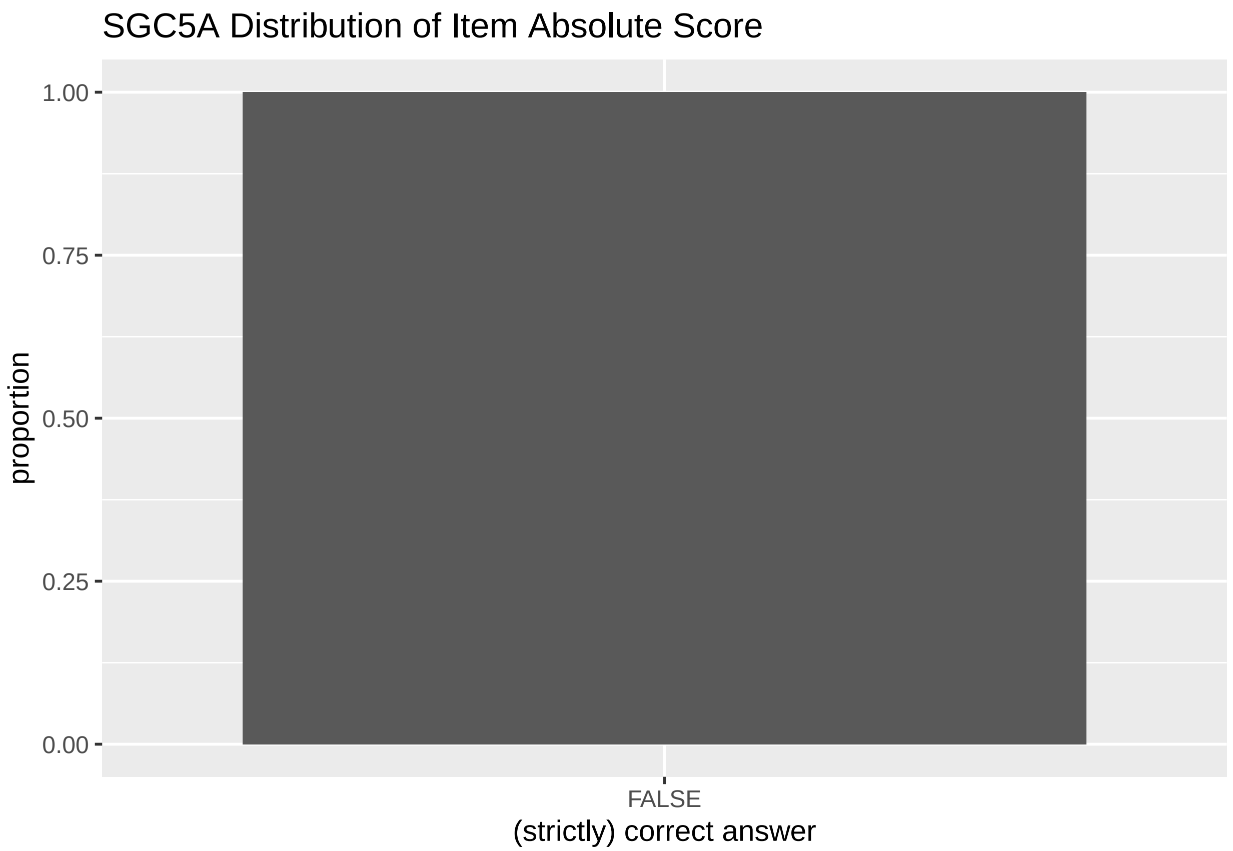
```
gf_histogram( ~absolute_score ,data = df_participants) +
  labs(title = "SGC5A Distribution of Absolute Score")
```



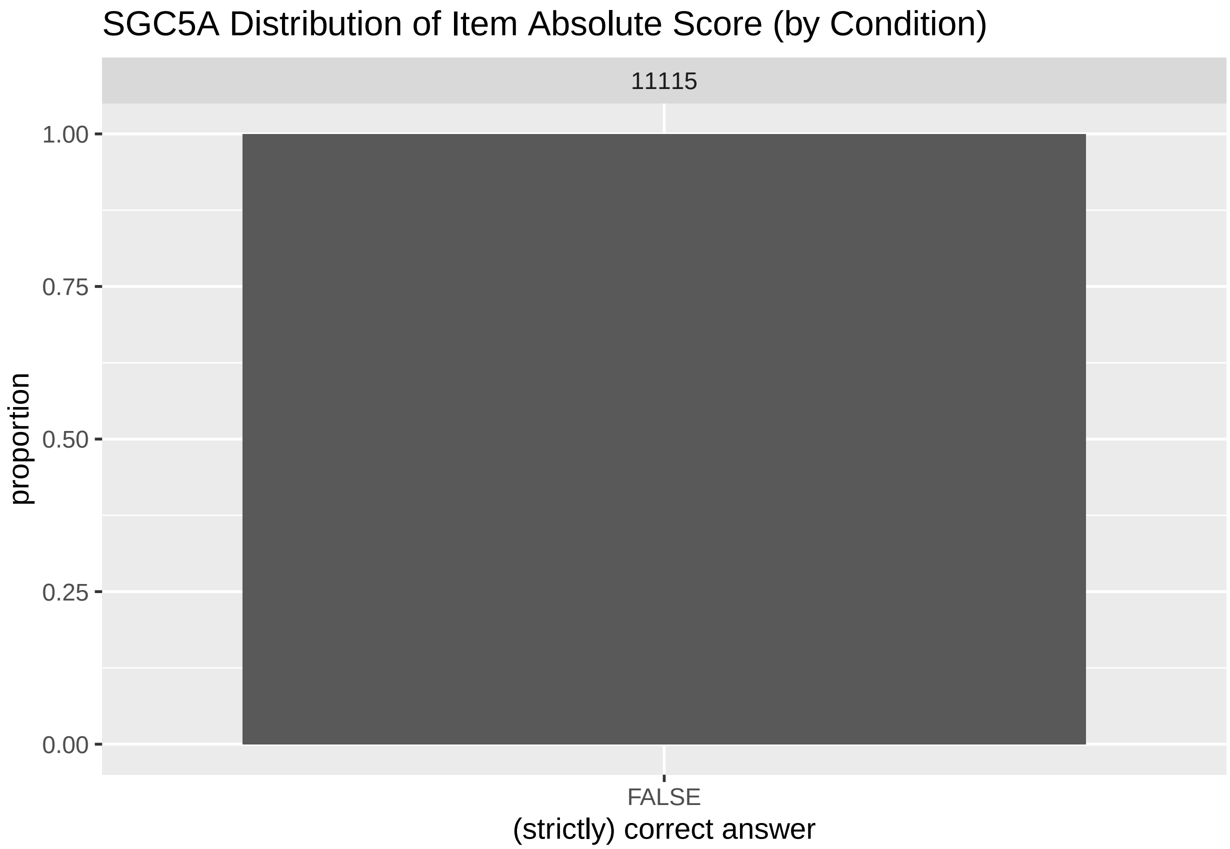
```
gf_histogram( ~absolute_score ,data = df_participants) %>%
  gf_facet_wrap(~pretty_condition) +
  labs(title = "SGC5A Distribution of Absolute Score (by Condition)")
```



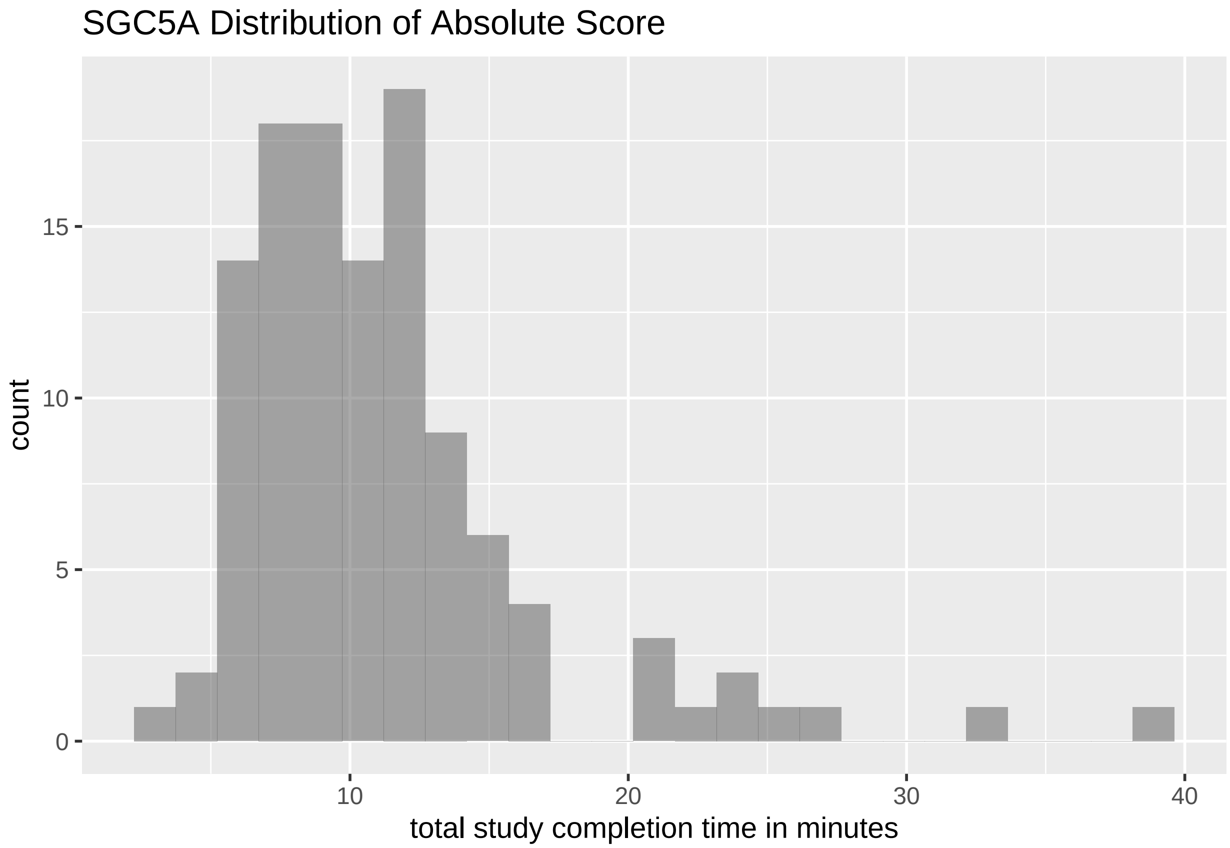
```
gf_props(~correct, data = df_items) +
  labs(title = "SGC5A Distribution of Item Absolute Score")
```



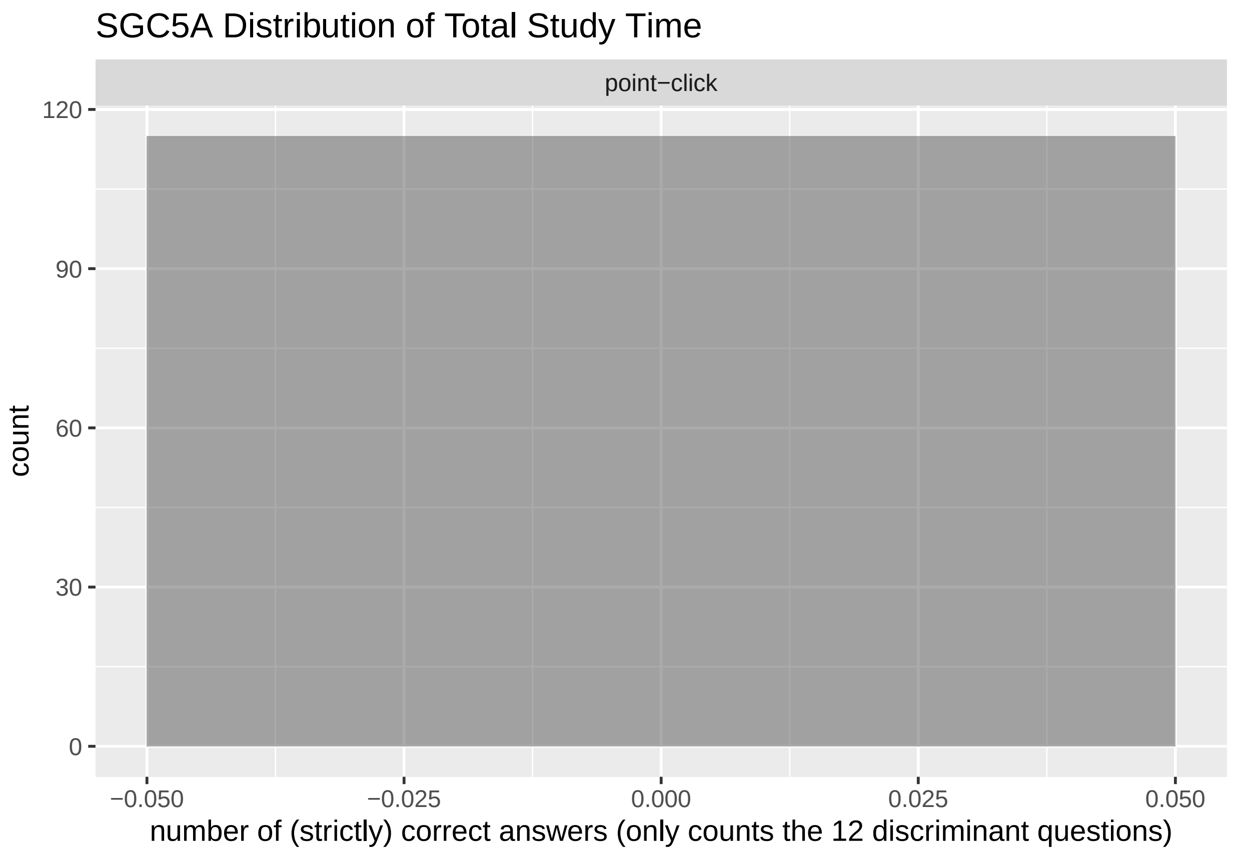
```
gf_props(~correct, data = df_items) %>%
  gf_facet_wrap(~condition) +
  labs(title = "SGC5A Distribution of Item Absolute Score (by Condition)")
```



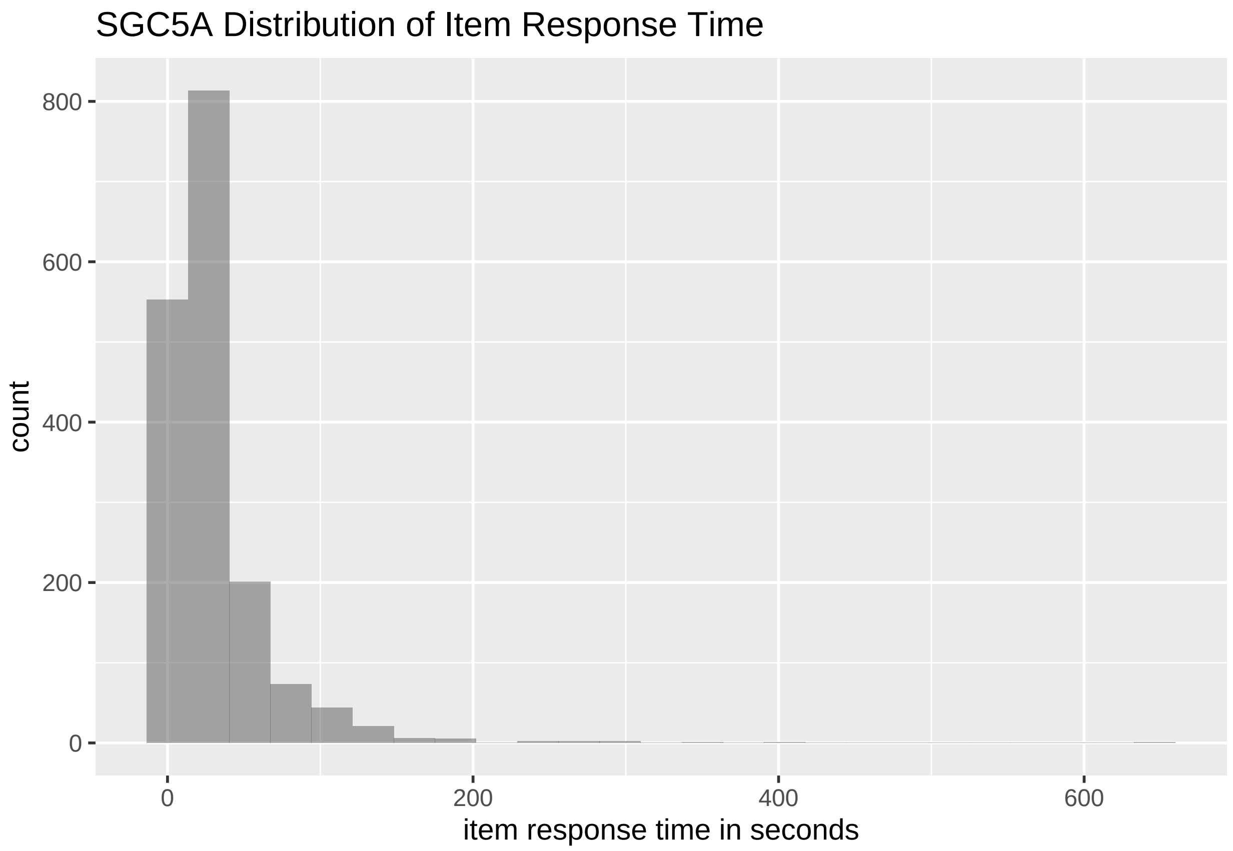
```
gf_histogram( ~totaltime_m ,data = df_participants) +
  labs(title = "SGC5A Distribution of Absolute Score")
```



```
gf_histogram( ~absolute_score ,data = df_participants) %>%
  gf_facet_wrap(~pretty_condition) +
  labs(title = "SGC5A Distribution of Total Study Time")
```



```
gf_histogram(~rt_s, data = df_items) +
  labs(title = "SGC5A Distribution of Item Response Time")
```



Data Export

Save Exclusions

For transparency, we save and identify the excluded data.

```r
write.csv(ex_participants,"output/excluded_participants_winter22_sgc5.csv", row.names = FALSE)
write.csv(ex_items,"output/excluded_items_winter22_sgc5.csv", row.names = FALSE)
```

Analysis-Ready Files

```r
#save csv files
write.csv(df_participants,"output/sgc5_participants.csv", row.names = FALSE)
write.csv(df_items,"output/sgc5_items.csv", row.names = FALSE)

 #export R DATA STRUCTURES (include codebook metadata)
rio::export(df_participants, "output/sgc5_participants.rds") # to R data structure file
rio::export(df_items, "output/sgc5_items.rds") # to R data structure file
```