E3 data wrangling

R. Cassano-Coleman

2025-06-26

This notebook combines and wrangles data for experiment 3.

Wrangle dataset 3A

Load dataset 3A and the musician/non-musician info.

```r
raw_3A <- read_csv('raw_combined_E3A.csv', show_col_types = FALSE)
groups_3A <- read_excel('../E3/sub_ids_3A.xlsx')
addl_subs <- read_excel('subs_additional_E3A.xlsx')
```

Keep only the main task.

```r
main_3A <- raw_3A %>%
  filter(grepl('OpenEndedSeg',Task_Name)) %>%
  # scramble level is in one of the following columns...
  mutate(scramble = ifelse(!is.na(scramble_cat_4), scramble_cat_4,
                           ifelse(!is.na(scramble_level), scramble_level,
                                  ifelse(!is.na(scramble_open_3), scramble_open_3, NA)))) %>%
  select(c(exp_subject_id, Trial_Id, Task_Name, participant_spacePress2, scramble)) %>%

  # add group info
  left_join(., groups_3A, by = join_by('exp_subject_id')) %>%
  relocate(Musician, .after = exp_subject_id) %>%
  # subjects who are left out did stimulus set 2, which is highly overlapping with stimulus set 1
  filter(!is.na(Musician))
```

Add the four additional subjects.

```r
main_3A <- addl_subs %>%
  select(colnames(main_3A)) %>%
  bind_rows(main_3A, .) %>%
  # capitalize Intact
  mutate(scramble = ifelse(scramble == 'intact', 'Intact', scramble))
```

Additional wrangling…

```r
main_3A %<>%
  # the number in Task_Name encodes stimulus set
  separate_wider_delim(Task_Name, delim = '_', names_sep = '_') %>% #, names = c(NULL, 'stimulus_set'))
  mutate(Task_Name_1 = NULL) %>%
  rename(stimulus_set = Task_Name_2) %>%

  # keep only the actual timestamps from `space_press2`
  # can't specify the column names in this step because there are some cells with NAs
  separate_wider_delim(participant_spacePress2, delim = '---', names_sep = '_') %>%
  mutate(participant_spacePress2_1 = NULL) %>%
  rename(responses = participant_spacePress2_2) %>%

  # transform response column into timestamp representation
  # first, replace any NA rows with 'values="' so all trials with no responses are consistent
  mutate(responses = ifelse(is.na(responses), 'values="', responses)) %>%
  # give each timestamp its own row
  separate_longer_delim(responses, delim = ';') %>%
  # some responses have values=X instead of values=;X
  separate_longer_delim(responses, delim = '=') %>%
  filter(responses != '') %>%
  filter(responses != '"') %>%

  # rename column for analysis
  rename(value = responses) %>%

  # move scramble
  relocate(scramble, .after = stimulus_set) %>%

  # rename Trial_Id to be stim_num to be consistent with 3B
  rename(stim_num = Trial_Id) %>%
  relocate(stim_num, .after = scramble)
```

Check for data completeness: every subject should have 2 trials per condition, but will have varying numbers of timestamps.

```r
main_3A %>%
  group_by(exp_subject_id) %>%
  summarize(conds = unique(scramble)) %>%
  summarize(n_conds = n())
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## `summarise()` has grouped output by 'exp_subject_id'. You can override using
## the `.groups` argument.

## # A tibble: 72 x 2
##    exp_subject_id n_conds
##             <dbl>   <int>
##  1         342236       4
##  2         342299       4
##  3         342301       4
##  4         344058       4
##  5         344111       4
##  6         344739       4
##  7         346025       4
##  8         352007       4
##  9         352021       4
## 10         353993       4
## # i 62 more rows
```

```
# R isn't happy but that's ok

# looking at it another way...
main_3A %>%
  group_by(exp_subject_id, scramble) %>%
  summarize(stimuli = unique(stim_num)) %>%
  group_by(exp_subject_id) %>%
  summarize(n_trials = n())
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'exp_subject_id', 'scramble'. You can
## override using the `.groups` argument.
```

```
## # A tibble: 72 x 2
##    exp_subject_id n_trials
##             <dbl>    <int>
##  1         342236        8
##  2         342299        8
##  3         342301        8
##  4         344058        8
##  5         344111        8
##  6         344739        8
##  7         346025        8
##  8         352007        8
##  9         352021        8
## 10         353993        8
## # i 62 more rows
```

At this point, every trial has an extra row (‘values="’), so subtract 1 from the calculation of rate.

```
rate_3A <- main_3A %>%
  group_by(exp_subject_id, Musician, stimulus_set, scramble, stim_num) %>%
  summarize(rate = ((n() - 1) / 64) * 60)
```

```
## `summarise()` has grouped output by 'exp_subject_id', 'Musician',
## 'stimulus_set', 'scramble'. You can override using the `.groups` argument.
```

Check for rates > 30

```
rate_3A %>%
  filter(rate > 30)
```

```
## # A tibble: 2 x 6
## # Groups:   exp_subject_id, Musician, stimulus_set, scramble [2]
##   exp_subject_id Musician stimulus_set scramble stim_num  rate
##            <dbl> <chr>    <chr>        <chr>       <dbl> <dbl>
## 1         344739 Yes      4            Intact          2  38.4
## 2         359121 No       4            8B              5  31.9
```

Average over all trials in a condition.

```
rate_3A %<>%
  group_by(exp_subject_id, Musician, stimulus_set, scramble) %>%
  summarize(mean_response_rate = mean(rate)) #%>%
```

```
## `summarise()` has grouped output by 'exp_subject_id', 'Musician',
## 'stimulus_set'. You can override using the `.groups` argument.
```

```
  #filter(mean_response_rate < 30)
```

Finish cleaning up those timestamps, including removing the extra rows

```
main_3A %<>%
  filter(value != 'values') %>%
  mutate(value = str_replace(value, '"', '')) %>%
  # convert from milliseconds to seconds
  mutate(value = as.numeric(value) / 1000)
```

Wrangle dataset 3B

Load dataset 3B.

```
raw_3B <- read_csv('raw_combined_E3B.csv', show_col_types = FALSE)
```

```
## New names:
## * `` -> `...36`
```

Select necessary columns and keep only main task.

```
main_3B <- raw_3B %>%
  # filter the main task
  # Block_Nr = 1 is before the main task: consent, instructions, headphone check, practice
  # Block_Nr = 3 is the post-survey
  filter(Block_Nr == '2') %>%

  # keep only the necessary columns
  # just want the responses from run 2, but add 'space_press1' here if you want to look at run 1 as wel
  select('Task_Name', 'space_press2', 'exp_subject_id') %>%
  # move subject ID to the front
  relocate('exp_subject_id') %>%

  # any task name with a percent in it is a "strategy" question
  filter(!grepl('%', Task_Name)) %>%

  # separate condition and stimulus number
  separate_wider_delim(Task_Name, delim = '_', names = c('scramble', 'stim_num')) %>%
  # capitalize Intact
  mutate(scramble = ifelse(scramble == 'intact', 'Intact', scramble)) %>%

  # keep only the actual timestamps from `space_press2`
  # can't specify the column names in this step because there are some cells with NAs
  separate_wider_delim(space_press2, delim = '---', names_sep = '_') %>%
  mutate(space_press2_1 = NULL) %>%
  rename(responses = space_press2_2) #%>%

  # before cleaning up this response column, check that each subject has complete data
  #group_by(exp_subject_id) %>%
  #summarize(count = n())
  # all of the subject have 12 trials (complete), except for 377660 who isn't included in the subsettin
```

Subset dataset 3B to match musician criteria and stimulus length of dataset 3A.

```
# list of subject IDs that match the musician criteria (either >5 years of experience or none at all)
# musicians (n = 13)
mus_sub_ids <- c(
  '377647', '393320', '393267', '393245', '393246',
  '393242', '393241', '393240', '393230', '393254',
  '393252', '393249', '393239')

# non-musicians (n = 10)
non_mus_sub_ids <- c(
  '377777', '377770', '377747', '377708', '377701',
  '377692', '377665', '377664', '377663', '393253')

subset_3B <- main_3B %>%
  # musician criteria
  mutate(Musician = ifelse(exp_subject_id %in% mus_sub_ids, 'Yes',
                         ifelse(exp_subject_id %in% non_mus_sub_ids, 'No', NA))) %>%
  relocate(Musician, .after = exp_subject_id) %>%
  filter(!is.na(Musician)) %>%

  # remove stimulus 2
  filter(stim_num != '2')
```

```
zeroRate_3B <- subset_3B %>%
  mutate(responses = ifelse(is.na(responses), 'values="', responses)) %>%
  filter(responses == 'values="') %>%
  mutate(rate = 0,
         responses = NULL) %>%

  # add stimulus set = 1 for when we combine
  mutate(stimulus_set = 1, .after = Musician)
```

```
subset_3B %<>%
  # transform response column into timestamp representation
  # first, remove NA rows
  filter(!is.na(responses)) %>%
  # replace 'values=' and " with an empty string
  mutate(responses = str_replace(responses, 'values=', '')) %>%
  mutate(responses = str_replace(responses, '"', '')) %>%
  # give each timestamp its own row
  separate_longer_delim(responses, delim = ';') %>%
  filter(responses != '') %>%
  # convert from milliseconds to seconds
  mutate(responses = as.numeric(responses) / 1000) %>%

  # subsetting rules:
  # when stim_num == 1, take all timestamps under 66 seconds
  filter(!(stim_num == '1' & responses > 66)) %>%
  # when stim_num == 3, take the last 66 seconds
  # everything after 16 seconds for Intact (since Intact is 41m/82s)
  filter(!(stim_num == '3' & scramble == 'Intact' & responses < 16)) %>%
  # everything after 10 seconds for Intact (since 8B is 38m/76s)
  filter(!(stim_num == '3' & scramble == '8B' & responses < 10)) %>%
  # everything after 14 seconds for 2B (since 2B is 40m/80s)
  filter(!(stim_num == '3' & scramble == '2B' & responses < 14)) %>%
  # everything after 14 seconds for 1B (since 1B is 40m/80s)
  filter(!(stim_num == '3' & scramble == '1B' & responses < 14)) %>%

  # adjust timing for stim_num == 3 so it starts at 0 for all conditions
  mutate(responses = ifelse(stim_num == 3, ifelse(
    scramble == 'Intact', responses - 16, ifelse(
      scramble == '8B', responses - 10, responses - 14
      )), responses)) %>%
  # check that nothing is below zero
  #filter(responses < 0)

  # rename column for analysis
  rename(value = responses) %>%

  # add stimulus set = 1 for when we combine
  mutate(stimulus_set = 1, .after = Musician)
```

Compute rate and exclude any cases where rate is greater than 1 response every 2 seconds (30 responses per minute).

```
# each of these stimuli are 66 seconds, rate is expressed per minute
rate_3B <- subset_3B %>%
  group_by(exp_subject_id, Musician, stimulus_set, scramble, stim_num) %>%
  summarize(rate = (n() / 66) * 60) %>%
  bind_rows(., zeroRate_3B)
```

```
## `summarise()` has grouped output by 'exp_subject_id', 'Musician',
## 'stimulus_set', 'scramble'. You can override using the `.groups` argument.
```

```
rate_3B %<>%
  # average over both runs for each subject
  group_by(exp_subject_id, Musician, stimulus_set, scramble) %>%
  summarize(mean_response_rate = mean(rate))
```

```
## `summarise()` has grouped output by 'exp_subject_id', 'Musician',
## 'stimulus_set'. You can override using the `.groups` argument.
```

None of these cases have rates > 30.

Combine

Rate

```
rate_3A$stimulus_set <- as.numeric(rate_3A$stimulus_set) # to match 3B
rate_combined <- bind_rows(rate_3A, rate_3B)
```

Check for any NAs

```
rate_combined %>% group_by(exp_subject_id) %>% summarize(count = n())
```

```
## # A tibble: 95 x 2
##    exp_subject_id count
##                <dbl> <int>
##   1         342236     4
##   2         342299     4
##   3         342301     4
##   4         344058     4
##   5         344111     4
##   6         344739     4
##   7         346025     4
##   8         352007     4
##   9         352021     4
## 10         353993     4
## # i 85 more rows
```

No missing data!

Make things into factors so we don't have to do it later.

```
rate_combined %<>%
  mutate(exp_subject_id = factor(exp_subject_id),
         Musician = factor(Musician, levels = c('Yes', 'No')),
         stimulus_set = factor(stimulus_set),
         scramble = factor(scramble, levels = c('Intact', '8B', '2B', '1B'))
         )
```

```
write_csv(rate_combined, '../E3/data/response_rate_by_sub.csv')
```

Timestamps

```
# make numeric things numeric...
main_3A$stimulus_set <- as.numeric(main_3A$stimulus_set) # to match 3B
subset_3B$stim_num <- as.numeric(subset_3B$stim_num) # to match 3A
# combine
timestamps_combined <- bind_rows(main_3A, subset_3B)
```

```
write_csv(timestamps_combined, '../E3/data/timestamps_filtered_long.csv')
```