Data Cleaning

26820228

10/08/2021

Setup

```
knitr::opts_chunk$set(echo = TRUE)
require("knitr")
```

```
## Loading required package: knitr
```

```
opts_knit$set(root.dir = "~/Library/Mobile Documents/com~apple~CloudDocs/Documents/Uni/Masters/Empirical Project/Code/Empirical_Project")

# turn off scientific notation
options(scipen = 999)
```

Load Libraries

```
library("dplyr") # for mutate function
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("reshape2") # for transforming data into wide and long format
library("tidyverse") # for tidying data
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.3     ✓ stringr 1.4.0
## ✓ tidyr   1.1.3     ✓ forcats 0.5.1
## ✓ readr   2.0.0

## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library("readr") # to read in data
```

Set Working Directory

```r
# please change this to your own working directory path
setwd("~/Library/Mobile Documents/com~apple~CloudDocs/Documents/Uni/Masters/Empirical Project/Code/Empirical_Project")
```

Fixation Data

Read in Fixation Data and Save Data to an Object

```r
# please change this to however you have stored the data file
df_fixations <- read.csv(file = "data/fixation_report.csv", header = TRUE, na.strings = ".")
```

Tidy / Check Fixation Dataset

```r
# retain only necessary columns
df_fixations <- data.frame(df_fixations[ , c("RECORDING_SESSION_LABEL",
                                             "TRIAL_INDEX",  "trial", "ext",
                                             "CURRENT_FIX_START", "CURRENT_FIX_END",
                                             "CURRENT_FIX_DURATION")])

# rename columns for easier interpretation
colnames(df_fixations) <- c("id", "trial_number", "condition", "extinction",
                            "fix_start", "fix_end", "fix_duration")

# and re-arrange by participant number (first to last)
df_fixations <- df_fixations %>%
  arrange(id)

# exclude CS-US trials
df_fixations <- df_fixations[df_fixations$condition != 3, ]

# retain only fixations that begin within 4000ms
# as these will begin within the stimulus presentation period
df_fixations <- data.frame(df_fixations[df_fixations$fix_start < 4000, ])
```

Compute Fixation Duration Averages

```r
#### for conditions 4, 5, 6 & 7 (acquisition CS+ & CS-, extinction CS+ & CS-)
# get the mean fixation duration for each participant for each trial
df_fix_duration <- df_fixations %>%
  group_by(id, trial_number, condition) %>%
  summarise(mean_fix_duration = mean(fix_duration))

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

```r
# get the mean fixation duration of the average fixation duration for
# each participant trial for each condition
df_fix_duration <- df_fix_duration %>%
  group_by(id, condition) %>%
  summarise(mean_fix_condition = mean(mean_fix_duration))
```

```
## `summarise()` has grouped output by 'id'. You can override using the `.gro
ups` argument.
```

```r
# transform to wide format
df_fix_duration_wide <- pivot_wider(df_fix_duration,
                                    id_cols = c(id, condition),
                                    names_from = condition,
                                    values_from = mean_fix_condition)


# rename columns for easier interpretation
colnames(df_fix_duration_wide) <- c("id", "acq_csp_fix_duration",
                                          "acq_csm_fix_duration",
                                          "ext_csp_fix_duration",
                                          "ext_csm_fix_duration")


#### now for early (cond 8) and late (cond 9) extinction
# get the mean fixation duration for each participant for each trial
df_fix_duration_extinction <- df_fixations %>%
  filter(extinction == 8 | extinction == 9) %>%
  group_by(id, trial_number, condition, extinction) %>%
  summarise(mean_fix_duration_extinction = mean(fix_duration))
```

```
## `summarise()` has grouped output by 'id', 'trial_number', 'condition'. You
can override using the `.groups` argument.
```

```r
# get the mean fixation duration of the average fixation duration for
# each participant trial for each condition
df_fix_duration_extinction <- df_fix_duration_extinction %>%
  group_by(id, condition, extinction) %>%
  summarise(mean_fix_condition_extinction = mean(mean_fix_duration_extinction
))
```

```
## `summarise()` has grouped output by 'id', 'condition'. You can override us
ing the `.groups` argument.
```

```r
# transform to wide format
df_fix_duration_extinction_wide <- pivot_wider(df_fix_duration_extinction,
                                    id_cols = c(id, condition, extinction),
                                    names_from = c(condition,extinction),
                                    values_from = mean_fix_condition_extinction)


# rename the columns for easier interpretation
```

```r
colnames(df_fix_duration_extinction_wide) <- c("id", "e_ext_csp_fix_duration"
,
                                               "l_ext_csp_fix_duration",
                                               "e_ext_csm_fix_duration",
                                               "l_ext_csm_fix_duration")
```

Compute Fixation Count Averages

```r
#### for conditions 4, 5, 6 & 7 (acquisition CS+ & CS-, extinction CS+ & CS-)

# obtain fixation count per participants per trial
df_fix_count <- df_fixations %>%
  group_by(id, trial_number, condition) %>%
  summarise(fix_count = length(fix_start))
```

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

```r
# get the mean fixation count for each participant for each condition
df_fix_count <- df_fix_count %>%
  group_by(id, condition) %>%
  summarise(mean_fix_condition = mean(fix_count))
```

```
## `summarise()` has grouped output by 'id'. You can override using the `.gro
ups` argument.
```

```r
# transform to wide format
df_fix_count_wide <- pivot_wider(df_fix_count,
                                 id_cols = c(id, condition),
                                 names_from = condition,
                                 values_from = mean_fix_condition)


# rename columns for easier interpretation
colnames(df_fix_count_wide) <- c("id", "acq_csp_fix_count",
                                           "acq_csm_fix_count",
                                           "ext_csp_fix_count",
                                           "ext_csm_fix_count")


#### now for early (cond 8) and late (cond 9) extinction
# obtain fixation count for each participant for each trial
df_fix_count_extinction <- df_fixations %>%
  filter(extinction == 8 | extinction == 9) %>%
  group_by(id, trial_number, condition, extinction) %>%
  summarise(fix_count = length(fix_start))
```

```
## `summarise()` has grouped output by 'id', 'trial_number', 'condition'. You
can override using the `.groups` argument.
```

```r
# get the mean fixation count for each participant for each condition
df_fix_count_extinction <- df_fix_count_extinction %>%
  group_by(id, condition, extinction) %>%
  summarise(mean_fix_condition_extinction = mean(fix_count))

## `summarise()` has grouped output by 'id', 'condition'. You can override us
ing the `.groups` argument.

# transform to wide format
df_fix_count_extinction_wide <- pivot_wider(df_fix_count_extinction,
                                id_cols = c(id, condition, extinction),
                                names_from = c(condition,extinction),
                                values_from = mean_fix_condition_extinction)


# rename the columns for easier interpretation
colnames(df_fix_count_extinction_wide) <- c("id", "e_ext_csp_fix_count",
                                            "l_ext_csp_fix_count",
                                            "e_ext_csm_fix_count",
                                            "l_ext_csm_fix_count")
```

Saccade Data

Read in Saccade Data and Save Data to an Object

```r
# please change this to however you have stored the data file
df_saccades <- read.csv(file = "data/saccade_report.csv", header = TRUE, na.s
trings = ".")
```

Tidy / Check Dataset

```r
# retain only necessary columns
df_saccades <- data.frame(df_saccades[ , c("RECORDING_SESSION_LABEL",
                                           "TRIAL_INDEX",  "trial", "ext",
                                           "CURRENT_SAC_START_TIME",
                                         "CURRENT_SAC_END_TIME",
                                           "CURRENT_SAC_AMPLITUDE")])

# rename columns for easier interpretation
colnames(df_saccades) <- c("id", "trial_number", "condition", "extinction",
                           "sacc_start", "sacc_end", "sacc_amplitude")

# and re-arrange by participant number (first to last)
df_saccades <- df_saccades %>%
  arrange(id)

 # retain only saccades that begin within 4000ms
# as these will begin within the stimulus presentation period
df_saccades <- data.frame(df_saccades[df_saccades$sacc_start < 4000, ])
```

```r
# exclude CS-US trials
df_saccades <- df_saccades[df_saccades$condition != 3, ]
```

Compute Saccade Amplitude Averages

```r
#### for conditions 4, 5, 6 & 7 (acquisition CS+ & CS-, extinction CS+ & CS-)

# get the mean saccade amplitude for each participant for each trial
df_sacc_amplitude <- df_saccades %>%
  group_by(id, trial_number, condition) %>%
  summarise(mean_sacc_amplitude = mean(sacc_amplitude), na.rm = TRUE)

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

# get the mean saccade amplitude of the average saccade amplitude for
# each participant trial for each condition
df_sacc_amplitude <- df_sacc_amplitude %>%
  group_by(id, condition) %>%
  summarise(mean_sacc_amplitude_condition = mean(mean_sacc_amplitude), na.rm
= TRUE)

## `summarise()` has grouped output by 'id'. You can override using the `.gro
ups` argument.

# transform to wide format
df_sacc_amplitude_wide <- pivot_wider(df_sacc_amplitude,
                                      id_cols = c(id, condition),
                                      names_from = condition,
                                      values_from = mean_sacc_amplitude_condition)


# rename columns for easier interpretation
colnames(df_sacc_amplitude_wide) <- c("id", "acq_csp_sacc_amplitude",
                                      "acq_csm_sacc_amplitude",
                                      "ext_csp_sacc_amplitude",
                                      "ext_csm_sacc_amplitude")


#### now for early (cond 8) and late (cond 9) extinction
# get the mean saccade amplitude for each participant for each trial
df_sacc_amplitude_extinction <- df_saccades %>%
  filter(extinction == 8 | extinction == 9) %>%
  group_by(id, trial_number, condition, extinction) %>%
  summarise(mean_sacc_amplitude_extinction = mean(sacc_amplitude))

## `summarise()` has grouped output by 'id', 'trial_number', 'condition'. You
can override using the `.groups` argument.

# get the mean saccade amplitude of the average saccade amplitude for
# each participant trial for each condition
```

```
df_sacc_amplitude_extinction <- df_sacc_amplitude_extinction %>%
  group_by(id, condition, extinction) %>%
  summarise(mean_sacc_amplitude_condition_extinction = mean(mean_sacc_amplitu
de_extinction))

## `summarise()` has grouped output by 'id', 'condition'. You can override us
ing the `.groups` argument.

# transform to wide format
df_sacc_amplitude_extinction_wide <- pivot_wider(df_sacc_amplitude_extinction
,
                                id_cols = c(id, condition, extinction),
                                names_from = c(condition,extinction),
                                values_from = mean_sacc_amplitude_condition_e
xtinction)


# rename the columns for easier interpretation
colnames(df_sacc_amplitude_extinction_wide) <- c("id", "e_ext_csp_sacc_amplit
ude",
                                          "l_ext_csp_sacc_amplitude",
                                          "e_ext_csm_sacc_amplitude",
                                          "l_ext_csm_sacc_amplitude")
```

Questionnaire and Demographic Data

Read in Questionnaire Data and Save Data to an Object

```
# please change this to however you have stored the data file
df_questionnaires <- read.csv(file = "data/questionnaires.csv", header = TRUE
, na.strings = ".")
```

Read in Demographics Data and Save Data to an Object

```
# please change this to however you have stored the data file
df_demographics <- read.csv(file = "data/demographics.csv", header = TRUE, na
.strings = ".")
```

Code the Demographic Data

```
df_demographics <- df_demographics %>%
  mutate(sex = recode(sex,
                      "Female (Gender: Male)" = "Female")) %>%
  mutate(ethnicity = recode(ethnicity,
                            "Asian" = "Asian",
                            "Asian- Malaysian Chinese" = "Asian",
                            "Asian- Chinese" = "Asian",
                            "Chinese" = "Asian",
                            "Asian - Pakistani" = "Asian",
                            "Asian (Pakistani)" = "Asian",
                            "Pakistan" = "Asian",
```

```
                              "British Pakistani" = "Asian",
                              "Indian" = "Asian",
                              "Malay" = "Asian",
                              "SriLanken" = "Asian",
                              "British Asian (Indian)" = "Asian",
                              "Kazakh" = "Asian",
                              "Black" = "Black",
                              "Black African" = "Black",
                              "Middle Eastern/ Arab" = "Middle Eastern/ Arab",
                              "Arab- North African" = "Middle Eastern/ Arab",
                              "Arab" = "Middle Eastern/ Arab",
                              "Mixed White/ Asian" = "Mixed",
                              "White/ Asian" = "Mixed",
                              "Half British, Half Asian" = "NA",
                              "White" = "White",
                              "White- other" = "White",
                              "White- Caucasian" = "White",
                              "British White" = "White",
                              "Caucasian" = "White",
                              "White Other" = "White",
                              "White British" = "White",
                              "white" = "White",
                              "British (white)" = "White",
                              "British" = "NA",
                              "Bulgarian" = "NA",
                              "Greek Cypriot" = "NA",
                              "Greek" = "NA",
                              "Russian" = "NA")) %>%
  mutate(sexual_orientation = recode(sexual_orientation,
                                     "Heterosexual" = "Heterosexual",
                                     "Straight" = "Heterosexual",
                                     "Straight Woman" = "Heterosexual",
                                     "Straight/ Heterosexual" = "Heterosexual
",
                                     "Male" = "Heterosexual",
                                     "Gay" = "Sexual Minority",
                                     "Gay Man" = "Sexual Minority",
                                     "Bisexual" = "Sexual Minority",
                                     "Bisexual/Queer" = "Sexual Minority",
                                     "Lesbian" = "Sexual Minority",
                                     "Lesbian/ Queer" = "Sexual Minority",
                                     "Pansexual" = "Sexual Minority"))
```

Bind the Data

```
# bind the data (for all participants)
dataframe_1 <- cbind(df_fix_duration_wide, df_fix_duration_extinction_wide,
                     df_fix_count_wide, df_fix_count_extinction_wide,
                     df_sacc_amplitude_wide,df_sacc_amplitude_extinction_wide
```

```
,
                     df_questionnaires, df_demographics)

## New names:
## * id -> id...1
## * id -> id...6
## * id -> id...11
## * id -> id...16
## * id -> id...21
## * ...

# remove repeated participant id columns
dataframe_1 <- dataframe_1[ , -c(6, 11, 16, 21, 26, 31, 80)]

# rename columns again
colnames(dataframe_1)[1] <- "id"

# create subdf for all participants except 'definitely exclude'
dataframe_2 <- dataframe_1[-c(2, 74, 101, 102, 110), ]

# create subdf for good AOI data participants only
dataframe_3 <- dataframe_1[c(8, 9, 13, 14, 16, 29, 30, 32, 37, 38, 40, 45, 46
,
                             48, 51, 53, 63, 64, 68, 69, 79, 100, 104, 107,
                             108, 121, 129, 131), ]

# and write each to csv
write.csv(dataframe_1, "data/cleaned/dataframe_1.csv", row.names = FALSE)
write.csv(dataframe_2, "data/cleaned/dataframe_2.csv", row.names = FALSE)
write.csv(dataframe_3, "data/cleaned/dataframe_3.csv", row.names = FALSE)
```