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()

library("readr") # to read in data

### *Set Working Directory*

# 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*

# 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*

# 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*

#### 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.

# 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 `.groups` argument.

# 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.

# 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 using the `.groups` argument.

# 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  
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*

#### 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.

# 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 `.groups` argument.

# 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.

# 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 using 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*

# please change this to however you have stored the data file   
df\_saccades <- read.csv(file = "data/saccade\_report.csv", header = TRUE, na.strings = ".")

## *Tidy / Check Dataset*

# 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, ])  
  
# exclude CS-US trials  
df\_saccades <- df\_saccades[df\_saccades$condition != 3, ]

## *Compute Saccade Amplitude Averages*

#### 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 `.groups` 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\_amplitude\_extinction))

## `summarise()` has grouped output by 'id', 'condition'. You can override using 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\_extinction)  
  
  
# rename the columns for easier interpretation  
colnames(df\_sacc\_amplitude\_extinction\_wide) <- c("id", "e\_ext\_csp\_sacc\_amplitude",   
 "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)