Injustice Attention Report

Tom Beesley

Setup and loading in the data

The data I'm loading in here is the fixation data for all participants. This is after having processed the raw data into fixations, which is handled in another script (stitch_data.R)

```
library(tidyverse)
library(eyetools)
library(effectsize)

load("inj_unsw_data.RData")

eg_stim <-
  eg_stim %>%
  mutate(pNum = as.numeric(pNum))
```

Setting up AOIs

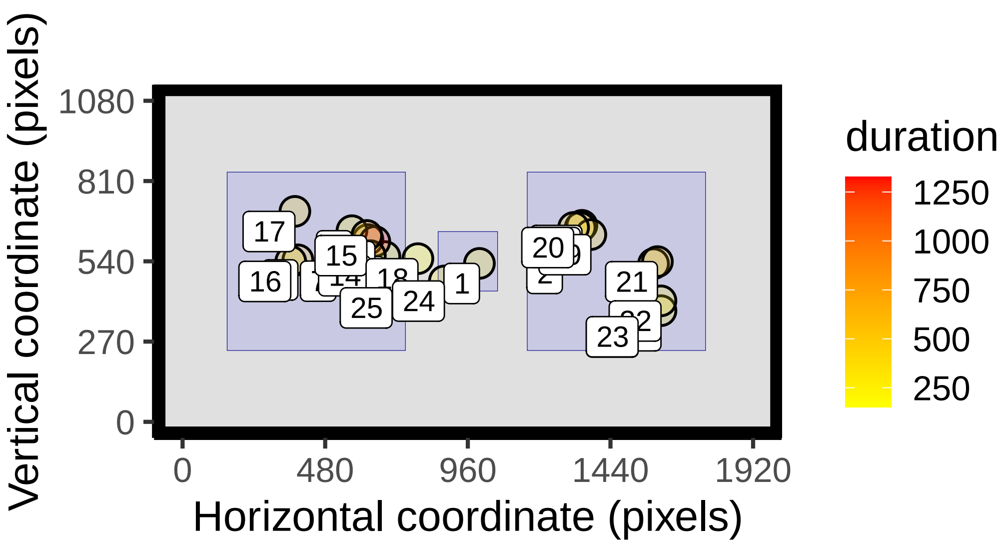
I set up three AOIs, for left, fixation, and right. Left and right are just guessed from the clustering of the fixations. We'll want to do something more precise with the actual placements of the stimuli further along the line.

```
# Set AOIs
AOI_stims <- data.frame(matrix(nrow = 3, ncol = 4))
colnames(AOI_stims) <- c("x", "y", "width_radius", "height")

AOI_stims[1,] <- c(450, 540, 600, 600) # X, Y, W, H - left
AOI_stims[2,] <- c(1460, 540, 600, 600) # X, Y, W, H - right
AOI_stims[3,] <- c(960, 540, 200, 200) # X, Y, W, H - fixation

AOI_stims_names <- c("left", "right", "centre")
```

Number of fixations

Here's what a typical trial looks like with 25 fixations (using `eyetools::spatial_plot()`):

```
data_trial <-
  eg_stim %>%
  filter(pNum == 001,
         trial == 3)

eyetools::spatial_plot(fix_data = data_trial,
                       AOIs = AOI_stims)
```



You can see from the index numbers that there is back and forth processing between the two stimuli. We can use `eyetools::AOI_seq()` to look at how the eye movements entered the different AOIs (here, just for the first 5 trials for this participant):

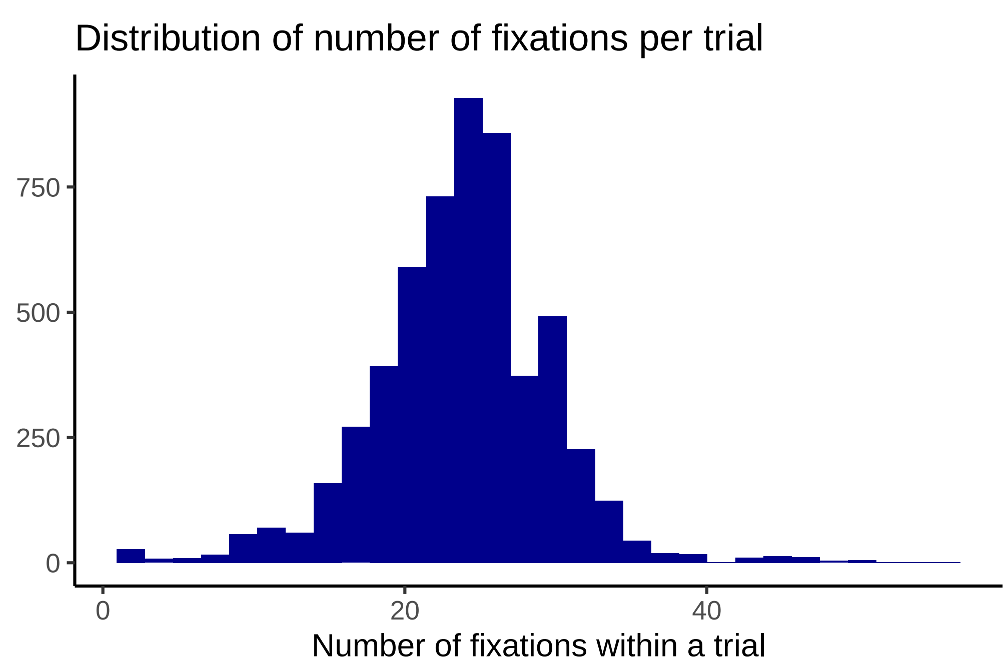
```
data_p1 <-
  eg_stim %>%
  filter(pNum == 001)

eyetools::AOI_seq(data = data_p1,
                  AOIs = AOI_stims,
```

```
                    AOI_names = AOI_stims_names) %>%
  head(5)
```

```
  trial                                      AOI_entry_seq
1     1 centre;right;centre;right;left;right;centre;right
2     2                centre;right;left;right;left;right
3     3    centre;right;left;right;left;right;centre;left
4     4                            centre;left;right;left
5     5          centre;left;right;left;centre;right;left
```

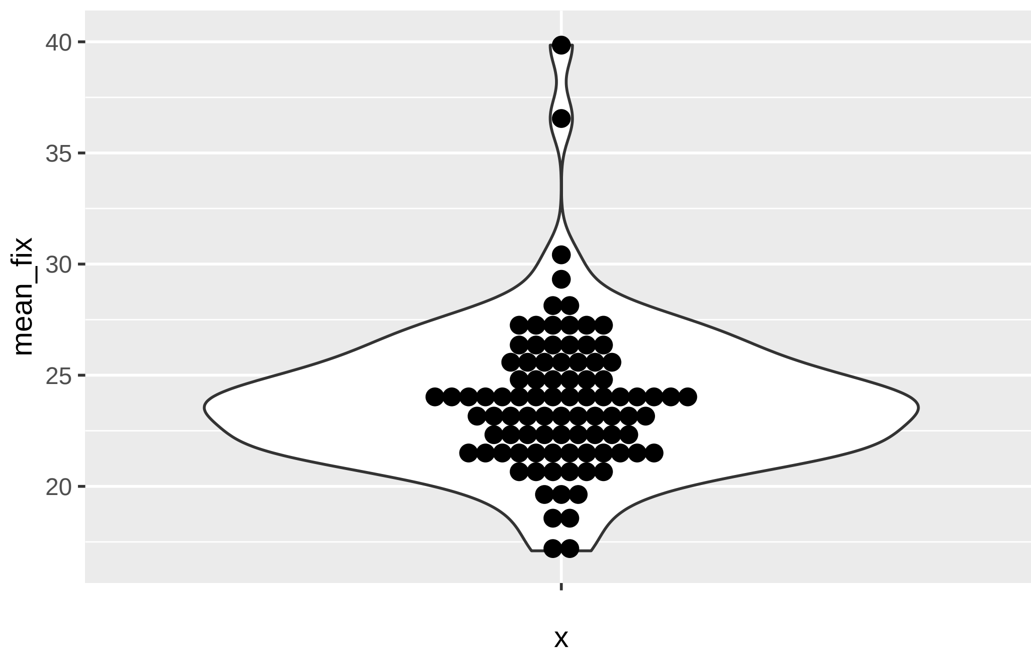
Here's the distribution of the number of fixations on a trial for all the data (for all participants):

```
eg_stim %>%
  group_by(pNum, trial) %>%
  summarise(nFix = n()) %>%
  ggplot() +
  geom_histogram(aes(nFix), bins = 30, fill = "dark blue") +
  theme_classic(base_size = 12) +
  labs(title = "Distribution of number of fixations per trial",
       x = "Number of fixations within a trial",
       y = "")
```



Do the participants vary much in the mean number of fixations per trial? There are a couple of outliers in terms of number of fixations.

```
eg_stim %>%
  group_by(pNum, trial) %>%
  summarise(n_fix = n()) %>%
  group_by(pNum) %>%
  summarise(mean_fix = mean(n_fix)) %>%
  ggplot(aes(mean_fix, x = "")) +
  geom_violin() +
  geom_dotplot(binaxis = "y", stackdir = "center")
```



Looking at time in AOIs for the different trial types / stimulus valence

Following that exploration, we can take a look at the more important stuff:

The following code takes the pair types that you've defined and matches them up with the trial data for each participant. Thus we end up with a dataframe that contains the time spent in each area of interest, and what was presented on the screen. Then depending on the location of the target (which I take to be the first stimulus named in each pair) we take the data from AOI_1 and AOI_3. This means that we then have columns that reflect time on "Stim 1" (the first stimulus in each pair) and “Stim 2” (the second stimulus in each pair), irrespective of whether they are on the left or right each trial.

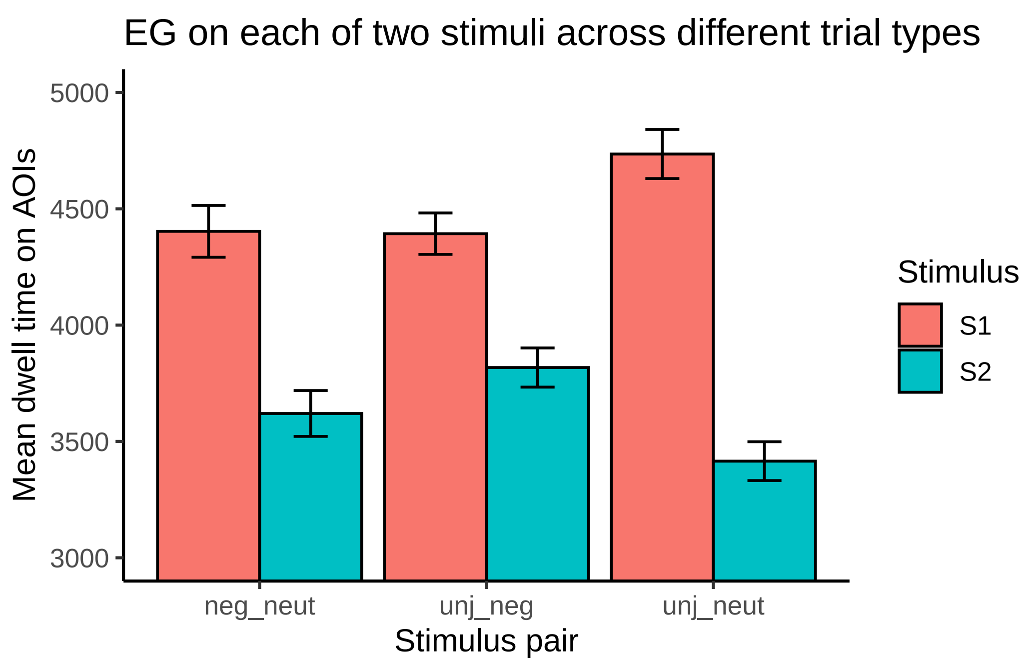
```
stim_codes <-
  read_csv("stim_details.csv") %>%
  rename(trial_ID = pair_num)


id_data <-
  id_data %>%
  left_join(stim_codes, by = "trial_ID")

eg_list <- split.data.frame(eg_stim, eg_stim[,1])

AOI_list <- lapply(eg_list, AOI_time, AOIs = AOI_stims)

AOI_df <- do.call(rbind.data.frame, AOI_list)

eg_data_combined <- cbind(id_data, AOI_df) # bind together the data

eg_data_combined <-
  eg_data_combined %>%
  mutate(across(AOI_1:AOI_3, ~as.numeric(.x)))

# pair type
# 1 = UNJ-NEG
# 2 = UNJ-NEUT
# 3 = NEG-NEUT

# target_loc
# 1 = LHS
# 2 = RHS

eg_data_full <-
  eg_data_combined %>%
  mutate(pair_type = case_match(pair_type,
                                1 ~ "unj_neg",
                                2 ~ "unj_neut",
                                3 ~ "neg_neut")) %>%
  mutate(stim_1_eg = case_when(target_loc == 1 ~ AOI_1,
                               target_loc == 2 ~ AOI_2),
         stim_2_eg = case_when(target_loc == 1 ~ AOI_2,
                               target_loc == 2 ~ AOI_1))
```

Summarising the data for visualisation

Taking this final combined dataframe, we can summarise how much time each person spends on each stimulus in the pair, as a function of the trial type. This is plotted below:

```
# summarise the data for each participant, to get
# the critical 6 values (3 pairs x 2 stimuli)
eg_means_summary <-
  eg_data_full %>%
  group_by(pNum, pair_type) %>%
  summarise(meanS1 = mean(stim_1_eg, na.rm = TRUE),
            meanS2 = mean(stim_2_eg, na.rm = TRUE))

# rather than have two columns for the two stimuli
# we should pivot the data and make stimulus a variable
eg_means_summary <-
eg_means_summary %>%
  pivot_longer(cols = c("meanS1", "meanS2"),
               names_prefix = "mean",
               names_to = "stimulus",
               values_to = "EG_time")

# calculate grand means and SE for the sample (for plotting)
eg_fig <-
  eg_means_summary %>%
  group_by(pair_type, stimulus) %>%
  summarise(meanEG = mean(EG_time),
            SE_EG = sd(EG_time)/sqrt(n()))

# plot the data
eg_fig %>%
  ggplot(aes(x = pair_type, y = meanEG,
             fill = stimulus,
             ymin = meanEG - SE_EG,
             ymax = meanEG + SE_EG)) +
  geom_col(color= "black",
           position=position_dodge()) +
  geom_errorbar(position = position_dodge(.9),
                width = .3) +
```

```
  coord_cartesian(ylim = c(3000, 5000)) +
  theme_classic(base_size = 12) +
  labs(y = "Mean dwell time on AOIs",
       x = "Stimulus pair",
       fill = "Stimulus",
       title = "EG on each of two stimuli across different trial types")
```



We'll run a t-test to see if the difference between unjust and negative stimuli is a real effect - yes it is! And with a medium effect size.

```
# filter to just the UNJ-NEG data
t_data_unj_neg <-
  eg_means_summary %>%
  filter(pair_type == "unj_neg")

# paired t-test
t.test(data = t_data_unj_neg,
       EG_time~stimulus,
       paired = TRUE)
```

```
    Paired t-test

data:  EG_time by stimulus
t = 6.1749, df = 91, p-value = 1.821e-08
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 390.2473 760.3882
sample estimates:
mean difference
       575.3177
```

```
# relevant effect size
effectsize::cohens_d(data = t_data_unj_neg,
                     EG_time~stimulus,
                     paired = TRUE)
```

```
Cohen's d |       95% CI
------------------------
0.64      | [0.42, 0.87]
```