E4-analysis-calibration

Rachel Ryskin

```
data.files <- list.files('data/run-2', full.names = TRUE)
data.tables <- lapply(data.files, function(file){
  data.table <- fromJSON(file)
  return(data.table)
})
all.data <- bind_rows(data.tables)
```

```
all.data.calib = all.data %>%
  filter(trial_type == "webgazer-validate") %>%
  dplyr::select(subject, trial_index,  percent_in_roi, average_offset) %>%
  tidyr::unnest(percent_in_roi)

summary.data.calib = all.data.calib %>%
  group_by(subject, trial_index) %>%
  summarize(mean_percent_in_roi = mean(percent_in_roi)) %>%
  group_by(subject) %>%
  mutate(calib_num = row_number())
```

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

```
ggplot(summary.data.calib)+
  geom_line(aes(x = calib_num, y = mean_percent_in_roi, color=subject))+
  theme_bw()+
  theme(legend.position = "none")
```

```
summary.data.calib.wide = summary.data.calib %>%
  select(-trial_index) %>%
  pivot_wider(id_cols = subject, names_from=calib_num, values_from = mean_percent_in_roi )

# correlation between initial and halfway calibration
ggplot(summary.data.calib.wide %>% filter(is.na(`3`)), aes(x = `1`, y = `2`))+
  geom_point()+
  geom_smooth(method = 'lm')+
  theme_bw()+
  theme(legend.position = "none")

# correlation between 2 successive calibration attempts
ggplot(summary.data.calib.wide %>% filter(!is.na(`3`)), aes(x = `1`, y = `2`))+
  geom_point()+
  geom_smooth(method = 'lm')+
  theme_bw()+
  theme(legend.position = "none")
```

```r
# correlation between second attempt calibration and halfway
ggplot(summary.data.calib.wide %>% filter(!is.na(`3`)), aes(x = `2`, y = `3`))+
  geom_point()+
  geom_smooth(method = 'lm')+
  theme_bw()+
  theme(legend.position = "none")
```

```r
calib.by.subj = summary.data.calib %>%
  group_by(subject) %>%
  summarize(mean_percent_in_roi = mean(mean_percent_in_roi))

eyetracking.effects.by.subj = read_rds( "output/E4_eye-tracking_data.rds") %>%
  rename("condition" = compatibility) %>%
  filter(time.window == "post-instrument-onset", condition != "filler" ) %>%
  group_by(condition, time.window, subject) %>%
  summarize(M = mean(prop.fixations.animal)) %>%
  pivot_wider(names_from = condition, values_from = M) %>%
  mutate(bias_effect = modifier - instrument) %>%
  left_join(calib.by.subj, by = "subject")
```

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

```r
ggplot(eyetracking.effects.by.subj, aes(x = mean_percent_in_roi, y = modifier))+
  geom_point()+
  geom_smooth(method = "lm")
```

```r
write_rds(calib.by.subj,"output/calib_by_subj.rds")
```

```r
m_calib = broom::tidy(cor.test(eyetracking.effects.by.subj$mean_percent_in_roi, eyetracking.effects.by.s
#m_calib
```

```r
m_calib_r = m_calib %>% pull(estimate )
```

```r
ggplot(eyetracking.effects.by.subj, aes(x = mean_percent_in_roi, y = bias_effect))+
  geom_point()+
  geom_smooth(method = "lm") +
  labs(x="Calibration Score", y="Bias Effect") +
  theme_bw()+
  theme(panel.grid.minor = element_blank()) +     theme(
    panel.grid.minor = element_blank(),
    axis.title = element_text(size=14),       # Axis titles
    axis.text = element_text(size=14),        # Axis labels
    legend.title = element_text(size=14),     # Legend title
    legend.text = element_text(size=14)       # Legend text
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Participants' calibration quality, measured as the mean percentage of fixations that landed within 200 pixels of the calibration point, varied substantially (between 2.22, 97.36%). The quality of a participant's calibration

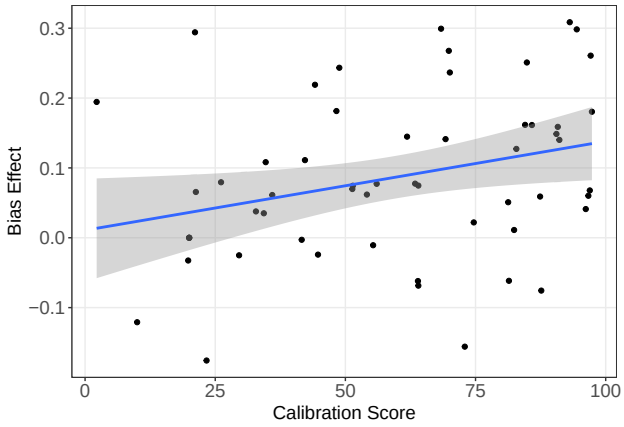

Figure 1: Calibration scores plotted against the verb bias effect (the difference in target animal fixations between modifier and instrument conditions).

significantly correlated with the participant's effect size (*Pearson's* r= 0.2947347, $p < 0.05$). The difference in target animal fixation proportions between modifier and instrument conditions was higher for participants with better calibration (see Figure @ref(fig:E4-calib-corr-plot)).

```
eyetracking.window1.summary.by.trial.orig = read_tsv("original_study_data/Experiment1_eye-tracking_verb
  mutate(subject = as.character(subj),
         prop.fixations.animal = TA/(TA+TI+CA+CI),
         #sum_dur = TA+TI+CA+CI,
         condition = factor(cond, levels = c(0, 2, 1, 9), labels = c("instrument", "equibiased", "modifi
         time.window = "post-verb-onset-pre-animal-onset",
         study = "original") %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal, study)
```

```
## Rows: 1944 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## dbl (8): TA, TI, CA, CI, subj, trialID, order, cond
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
eyetracking.window2.summary.by.trial.orig = read_tsv("original_study_data/Experiment1_eye-tracking_anima
  mutate(subject = as.character(subj),
         prop.fixations.animal = TA/(TA+TI+CA+CI),
         condition = factor(cond, levels = c(0, 2, 1, 9), labels = c("instrument", "equibiased", "modifi
         time.window = "post-animal-onset-pre-instrument-onset",
         study = "original") %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal, study)
```

```
## Rows: 1944 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## dbl (8): TA, TI, CA, CI, subj, trialID, order, cond
##
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
eyetracking.window3.summary.by.trial.orig = read_tsv("original_study_data/Experiment1_eye-tracking_instr
  mutate(subject = as.character(subj),
         prop.fixations.animal = TA/(TA+TI+CA+CI),
         condition = factor(cond, levels = c(0, 2, 1, 9), labels = c("instrument", "equibiased", "modifi
         time.window = "post-instrument-onset",
         study = "original") %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal, study)
```

```
## Rows: 1944 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## dbl (8): TA, TI, CA, CI, subj, trialID, order, cond
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
eyetracking.window.summary.by.trial = read_rds("output/E4_eye-tracking_data.rds")
```

```
eyetracking.window.summary.by.trial.post.exclusion = eyetracking.window.summary.by.trial %>%
  rename("condition" = compatibility) %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal) %>%
  mutate(study = "web (calib. > 50%)") %>%
  left_join(calib.by.subj, by = "subject") %>%
 filter(mean_percent_in_roi >= 50)


eyetracking.window.summary.by.subj.both.post.exclusion = bind_rows(eyetracking.window1.summary.by.trial
   filter(time.window != "end", time.window != "pre-verb-onset", condition != "filler" ) %>%
  group_by(study, condition, time.window, subject) %>%
  summarize(mean_prop = mean(prop.fixations.animal, na.rm = T)) %>%
  mutate(time.window = factor(time.window,
                              levels = c("post-verb-onset-pre-animal-onset",
                                         "post-animal-onset-pre-instrument-onset",
                                         "post-instrument-onset"),
                              labels = c('Verb to Animal \n("Rub the")',
                                         'Animal to Instr.\n("frog with the")',
                                         'Instr.+1.5s\n(feather...")')),
         condition  = factor(condition,
                             levels = c("modifier", "equibiased", "instrument"),
                             labels = c("Modifier", "Equi-biased", "Instrument"))
         ) %>%
  ungroup()

eyetracking.window.summary.both.post.exclusion = eyetracking.window.summary.by.subj.both.post.exclusion
  group_by(study, condition, time.window) %>%
  summarize(mean_prop = mean(mean_prop, na.rm = T),
            se_prop = sd(mean_prop, na.rm = T)/sqrt(n()))
```

```
ggplot(eyetracking.window.summary.by.subj.both.post.exclusion)+
  # stat_summary(data = eyetracking.window.summary.by.subj.both,
```

```
  #              aes(x=time.window, color=condition, y = mean_prop, shape = study),
  #              geom = 'point', fun = 'mean', size = 3)+
  stat_summary(aes(x=time.window, color=condition, y = mean_prop, shape = study),
               geom = 'point', fun = 'mean', size = 3)+
 stat_summary(aes(x=as.numeric(time.window), y = mean_prop, linetype = study, group=paste(condition, st
              geom = 'line', fun = 'mean', alpha = 0.5, color = "grey")+
  stat_summary(aes(x=as.numeric(time.window), y = mean_prop, color = condition, group = paste(condition
                geom = 'errorbar', fun.data = 'mean_cl_boot', width = 0.05)+
  #facet_wrap(~time.window)+
  scale_color_brewer(palette = "Set1")+
  scale_shape_manual(values = c(1,16))+
  scale_linetype_manual(values = c("dashed","solid"))+
  coord_cartesian(ylim = c(0,.8))+
  theme_classic()+
  labs(y = "Proportion of fixations", x = NULL) +     theme(
    panel.grid.minor = element_blank(),
    axis.title = element_text(size=14),       # Axis titles
    axis.text = element_text(size=14),        # Axis labels
    legend.title = element_text(size=14),     # Legend title
    legend.text = element_text(size=14)       # Legend text
  )
```

```
eyetracking.window.summary.by.trial.all = eyetracking.window.summary.by.trial %>%
  rename("condition" = compatibility) %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal) %>%
  mutate(study = "web") %>%
  left_join(calib.by.subj, by = "subject")
```

```
## Adding missing grouping variables: ‘sound‘
```

```
eyetracking.window.summary.by.trial.good.calib = eyetracking.window.summary.by.trial.all %>%
 filter(mean_percent_in_roi >= 50) %>%
  mutate(study = "web (calib. > 50%)")


eyetracking.window.summary.by.subj.three = bind_rows(eyetracking.window1.summary.by.trial.orig, eyetrac
   filter(time.window != "end", time.window != "pre-verb-onset", condition != "filler" ) %>%
  group_by(study, condition, time.window, subject) %>%
  summarize(mean_prop = mean(prop.fixations.animal, na.rm = T)) %>%
  mutate(time.window = factor(time.window,
                              levels = c("post-verb-onset-pre-animal-onset",
                                         "post-animal-onset-pre-instrument-onset",
                                         "post-instrument-onset"),
                              labels = c('Verb to Anim. \n("Rub the")',
                                         'Anim, to Instr.\n("frog with the")',
                                         'Instr.+1.5s\n(feather...")')),
         condition  = factor(condition,
                             levels = c("modifier", "equibiased", "instrument"),
                             labels = c("Modifier", "Equi-biased", "Instrument"))
         ) %>%
  ungroup()
```

```
## ‘summarise()‘ has grouped output by ’study’, ’condition’, ’time.window’. You
```

```
## can override using the `.groups` argument.
```

```
# eyetracking.window.summary.both.post.exclusion = eyetracking.window.summary.by.subj.both.post.exclusi
#   group_by(study, condition, time.window) %>%
#   summarize(mean_prop = mean(mean_prop, na.rm = T),
#             se_prop = sd(mean_prop, na.rm = T)/sqrt(n()))
```

```
ggplot(eyetracking.window.summary.by.subj.three)+
  # stat_summary(data = eyetracking.window.summary.by.subj.both,
  #              aes(x=time.window, color=condition, y = mean_prop, shape = study),
  #              geom = 'point', fun = 'mean', size = 3)+
  stat_summary(aes(x=as.numeric(time.window), color=condition, y = mean_prop, shape = study),
               geom = 'point', fun = 'mean', size = 3, position = position_dodge(width = 0))+
 stat_summary(aes(x=as.numeric(time.window), color = condition, y = mean_prop, linetype = study, group=
             geom = 'line', fun = 'mean', alpha = 0.5, position = position_dodge(width = 0))+
  stat_summary(aes(x=as.numeric(time.window), y = mean_prop, color = condition, group = interaction(cond
               geom = 'errorbar', fun.data = 'mean_cl_boot', width = 0.05, position = position_dodge(wi
  #facet_wrap(~time.window)+
  scale_color_brewer(palette = "Set1")+
  scale_shape_manual(values = c(1,16, 4))+
  scale_linetype_manual(values = c("dashed","solid", "dotted"))+
  coord_cartesian(ylim = c(0,.8))+
  theme_classic()+
  labs(y = "Proportion of fixations", x = NULL) +     theme(
    panel.grid.minor = element_blank(),
    axis.title = element_text(size=14),      # Axis titles
    axis.text = element_text(size=13),       # Axis labels
    legend.title = element_text(size=14),    # Legend title
    legend.text = element_text(size=14)      # Legend text
  )
```

```
eyetracking.window.3 = eyetracking.window.summary.by.trial %>%
  filter(time.window == "post-instrument-onset", compatibility != "filler" ) %>%
  mutate(condition = factor(compatibility, levels = c('instrument', 'equibiased', 'modifier'))) %>%
  left_join(calib.by.subj, by = "subject")
```

```
# Add orthogonal contrasts to model
contrasts(eyetracking.window.3$condition) <- cbind(c(-2/3, 1/3, 1/3), c(0, -1/2, 1/2))
```

```
model.time.window.3 <- lmer(prop.fixations.animal ~ condition + (1 +condition | subject) + (1 | trialID)
                            control = lmerControl(optimizer = "bobyqa",
                                                  optCtrl = list(maxfun = 2e6)))
summary(model.time.window.3)
```

```
eyetracking.window.3.good.calib = eyetracking.window.3 %>%
  filter(mean_percent_in_roi >= 50)
```

```
# Add orthogonal contrasts to model
contrasts(eyetracking.window.3.good.calib$condition) <- cbind(c(-2/3, 1/3, 1/3), c(0, -1/2, 1/2))
```

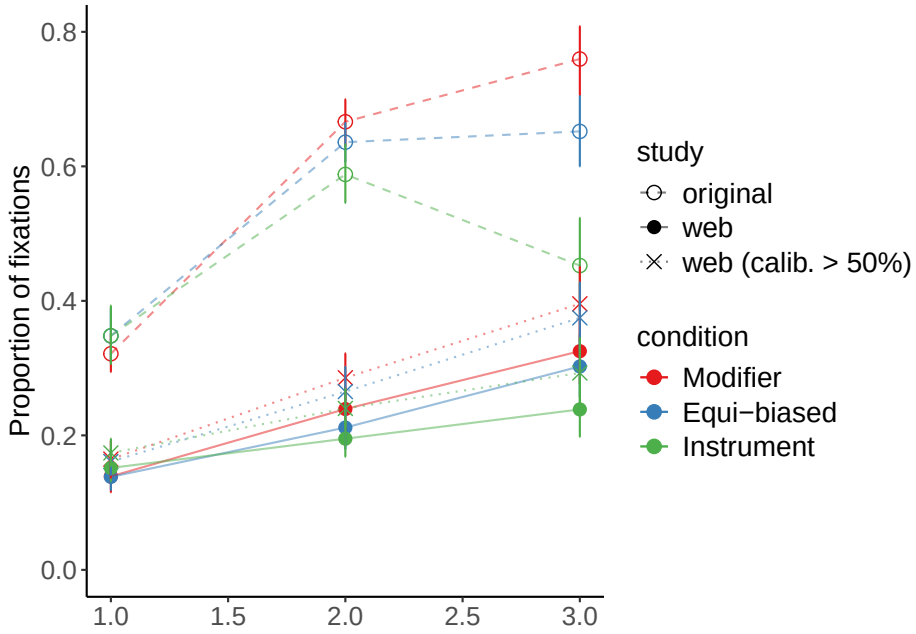

Figure 2: Proportion of target fixations by verb bias in the original dataset (Ryskin et al., 2017) and the current data collected online for the subset of participants with calibration quality >50%. Error bars reflect bootstrapped 95% CIs over subject means.

```
m.good.calib <- lmer(prop.fixations.animal ~ condition + (1  | subject) + (1 | trialID), data=eyetracki
                     control = lmerControl(optimizer = "bobyqa",
                                           optCtrl = list(maxfun = 2e6)))
#summary(m.good.calib)

m_calib_good = broom.mixed::tidy(m.good.calib)

m_calib_good_c1 = m_calib_good %>% filter(term == "condition1")
m_calib_good_c2 = m_calib_good %>% filter(term == "condition2")
```

Re-analysis After Exclusions A subset of 35 participants had calibration quality >50%. Figure @ref(fig:E4-proportion-fix-by-window-both-post-exclusion) shows proportions of fixations to the target animal in the web version for this subset alongside the full dataset, and the original data. Replicating the linear mixed-effects analysis (in the post-instrument onset time window only) on this subset of participants suggests that the effect of verb bias condition was larger in this subset than in the full dataset. Participants' preference to the target animal relative to the target instrument in the modifier-biased condition and the equi-biased conditions was greater than in the instrument-biased condition ($b = 0.1$, $SE = 0.02$, $p < 0.001$), but the difference betwen the modifier biased condition and the equi-biased condition was not significant ($b = 0.02$, $SE = 0.02$, $p = 0.29$).

```
eyetracking.window.3.great.calib = eyetracking.window.3 %>%
  filter(mean_percent_in_roi >= 75)
```

```
# Add orthogonal contrasts to model
contrasts(eyetracking.window.3.great.calib$condition) <- cbind(c(-2/3, 1/3, 1/3), c(0, -1/2, 1/2))
```

```r
m.great.calib <- lmer(prop.fixations.animal ~ condition + (1  | subject) + (1 | trialID), data=eyetracki
                            control = lmerControl(optimizer = "bobyqa",
                                                  optCtrl = list(maxfun = 2e6)))
#summary(m.great.calib)
m_calib_great = broom.mixed::tidy(m.great.calib)

m_calib_great_c1 = m_calib_great %>% filter(term == "condition1")
m_calib_great_c2 = m_calib_great %>% filter(term == "condition2")
```