E4-analysis-comparison-to-original

Rachel Ryskin

```
first.move = read_csv("output/E4_mouse_data.csv")
```

```
## Rows: 3078 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): subject, object, type, compatibility, target_instrument, target_ani...
## dbl (5): trialID, which_loc, instrument_loc, animal_loc, is.mouse.instrument
## lgl (1): images
##
## 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_csv( "output/E4_eye-tracking_data.csv") %>%
  rename("condition" = compatibility) %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal) %>%
  mutate(study = "web")
```

```
## Rows: 14667 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): subject, sound, compatibility, time.window
## dbl (3): trialID, prop.fixations.animal, prop.fixations.instrument
##
## 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.
```

```
first.move.orig = read_tsv("original_study_data/Experiment1_clickData.txt")
```

```
## Rows: 1296 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (1): condition
## dbl (6): subject, randomlyGeneratedNumber, order, trialID, clickedTargetAnim...
##
## 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.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", "modif
         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", "modif
         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_inst
  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", "modif
         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.
```

```
# Summarize the data by subject, calculating proportion of trials the the first move was to the animal,
first.move.web.subject.summary <- first.move %>%
  group_by(subject, compatibility) %>%
  summarize(prop.animal = mean(first.move.type == 'animal'),
            prop.instrument = mean(first.move.type == 'instrument'),
            prop.other = mean(first.move.type == 'other')) %>%
  pivot_longer(c('prop.animal', 'prop.instrument', 'prop.other'), names_to="target_type", values_to="pr
## ‘summarise()‘ has grouped output by ’subject’. You can override using the
## ‘.groups‘ argument.
```

```
# Summarize the condition-level data for a barplot.
first.move.web.summary <- first.move.web.subject.summary %>%
  group_by(compatibility, target_type) %>%
  summarize(M=mean(proportion), SE=sd(proportion)/sqrt(n())) %>%
  mutate(bias = factor(compatibility, levels = c("instrument", "equibiased", "modifier") ),
         study = "web")
```

```
## ‘summarise()‘ has grouped output by ’compatibility’. You can override using the
## ‘.groups‘ argument.
```

```
# Summarize the data by subject, calculating proportion of trials the the first move was to the animal,
first.move.orig.subject.summary <- first.move.orig %>%
  group_by(subject, condition) %>%
  summarize(prop.animal = mean(clickedTargetAnimalFirst),
            prop.instrument = mean(clickedTargetInstrumentFirst),
            prop.other = mean(clickedTargetAnimalFirst == 0 & clickedTargetInstrumentFirst == 0 )) %>%
  pivot_longer(c('prop.animal', 'prop.instrument', 'prop.other'), names_to="target_type", values_to="pro
```

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

```
# Summarize the condition-level data for a barplot.
first.move.orig.summary <- first.move.orig.subject.summary %>%
  group_by(condition, target_type) %>%
  summarize(M=mean(proportion), SE=sd(proportion)/sqrt(n())) %>%
  mutate(bias = factor(condition, levels = c("Inst", "Equi", "mod"), labels = c("instrument", "equibiase
         study = "original")
```

```
## ‘summarise()‘ has grouped output by ’condition’. You can override using the
## ‘.groups‘ argument.
```

The web version of the study qualitatively replicates the action and eye-tracking results of the original dataset [@ryskinVerbBiasesAre2017]. The mouse click results from both studies are summarized in Figure @ref(fig:E4-mouse-moves-fig-web-and-orig). The quantitative patterns of clicks were similar to those observed in the original dataset, though for Instrument-biased verbs, clicks were closer to evenly split between the animal and the instrument relative to the in-lab study where they were very clearly biased toward the instrument.

```
ggplot(bind_rows(
  first.move.orig.summary %>% mutate(study_type = "Original"),
  first.move.web.summary %>% mutate(study_type = "Web")),
  aes(x=bias, fill=target_type, y=M, ymin=M-SE, ymax=M+SE)) +
  geom_col(position=position_dodge(width=0.9), color = "black") +
  geom_errorbar(position=position_dodge(width=0.9), width=0.2) +
  scale_fill_brewer(palette = "Set1", name = "Location", labels = c("Animal", "Instrument", "Other")) +
  coord_cartesian(ylim = c(0,1)) +
  theme_classic() +
  labs(y = "Proportion of first mouse movements", x = "Bias") +
  facet_wrap(~study_type, ncol = 1)  # This creates vertical panels
```

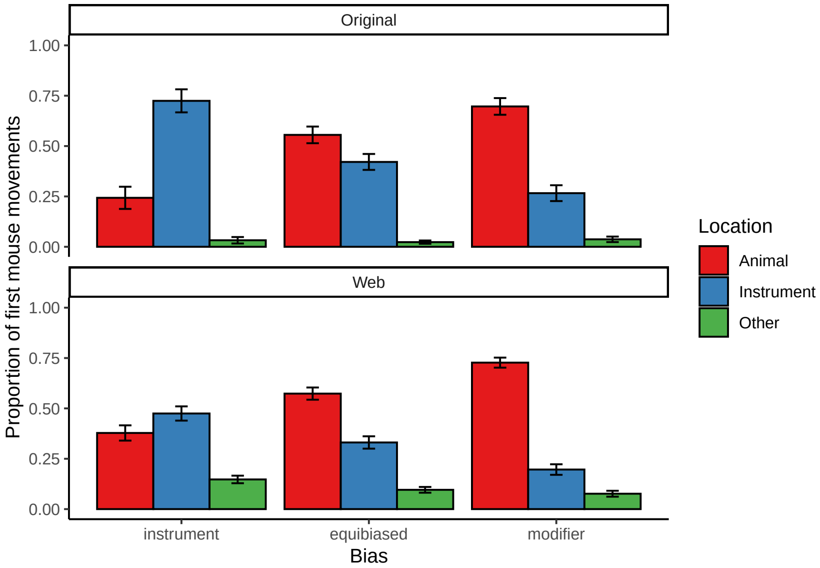

Figure 1: Proportion of first mouse movements by location and verb bias in the original dataset (Ryskin et al., 2017) and the current data collected online.

The eye-tracking results from both studies are summarized in Figure @ref(fig:E4-proportion-fix-by-window-both). For simplicity, and to reflect the dependent variable used in analyses, we average the proportion of fixations to the target animal within each time window. Though the qualitative patterns are replicated, proportions of fixations to the target animal were much lower in the web version of the study. This may reflect the fact that participants in the web study are less attentive and/or the quality of the webgazer eye-tracking system is lower, relative to the Eyelink 1000 which was used for the original study.

```
eyetracking.window.summary.by.trial.both = bind_rows(eyetracking.window1.summary.by.trial.orig, eyetrac
   filter(time.window != "end", time.window != "pre-verb-onset", condition != "filler" )
```

```
eyetracking.window.summary.by.subj.both = bind_rows(eyetracking.window1.summary.by.trial.orig, eyetracki
   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 onset\n("frog with the")',
                                       'Instrument+1500ms\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 = eyetracking.window.summary.by.subj.both %>%
  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()))
```

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

```
ggplot(eyetracking.window.summary.by.subj.both)+
  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, stu
             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
  )
```

```
## Warning: Computation failed in `stat_summary()`.
## Caused by error in `fun.data()`:
## ! The package "Hmisc" is required.
```

```
orig.data.time.window.1 <- eyetracking.window1.summary.by.trial.orig %>%
  filter(condition != "filler") %>%
  mutate(condition = fct_drop(condition))
contrasts(orig.data.time.window.1$condition) <- cbind(c(-2/3, 1/3, 1/3), c(0, -1/2, 1/2))

model.time.window.1 <- lmer(prop.fixations.animal ~ condition + (1 | subject) + (1 | trialID), data=orig
summary(model.time.window.1)
```

```
orig.data.time.window.2 <- eyetracking.window2.summary.by.trial.orig %>%
  filter(condition != "filler") %>%
  mutate(condition = fct_drop(condition))
contrasts(orig.data.time.window.2$condition) <- cbind(c(-2/3, 1/3, 1/3), c(0, -1/2, 1/2))

model.time.window.2 <- lmer(prop.fixations.animal ~ condition + (1 | subject) + (1 | trialID), data=orig
summary(model.time.window.2)
```

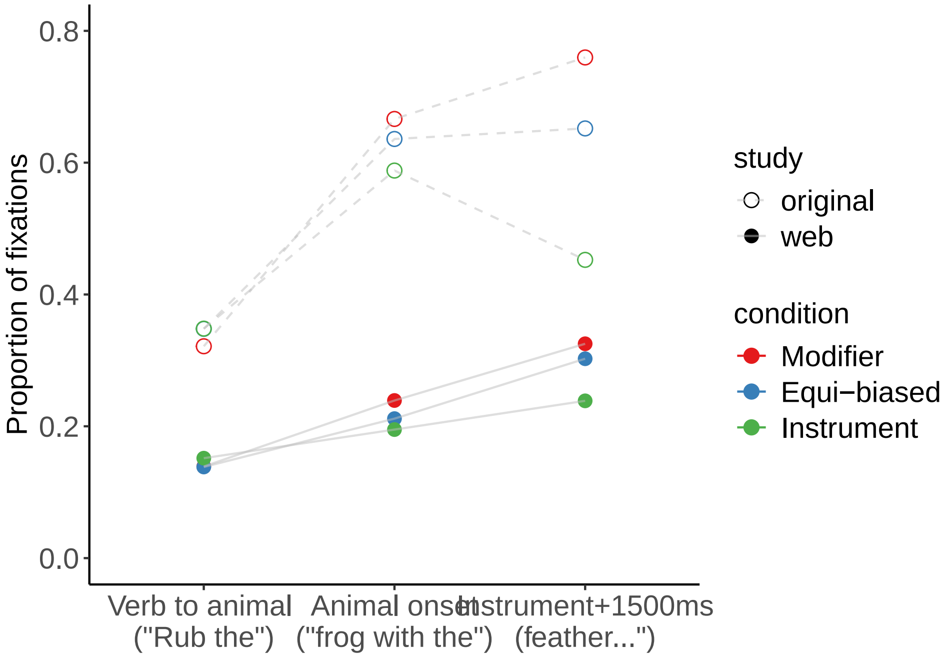

Figure 2: Proportion of target fixations by verb bias in the original dataset (Ryskin et al., 2017) and the current data collected online. Error bars reflect bootstrapped 95% CIs over subject means

```
orig.data.time.window.3 <- eyetracking.window3.summary.by.trial.orig %>%
  filter(condition != "filler") %>%
  mutate(condition = fct_drop(condition))
contrasts(orig.data.time.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 | subject) + (1 | trialID), data=orig
summary(model.time.window.3)
```