Group B Analysis

```r
library(jsonlite)
library(dplyr)
library(tidyr)
library(ggplot2)
library(patchwork)
library(lmerTest)
library(broom.mixed)
```

```r
data.files <- list.files('data', full.names = TRUE, pattern=".json")
data.tables <- lapply(data.files, function(file){
  data.table <- fromJSON(file)
  return(data.table)
})
all.data <- bind_rows(data.tables)
```

```r
task.data <- all.data %>%
  filter(!is.na(relation)) %>%
  select(subject_id, trial_index, stimulus, relation, quadrant, rt, correct_response, response, webgazer
  tidyr::unpack(webgazer_targets) %>%
  tidyr::unpack(`#screen`) %>%
  select(-top,-left,-x,-y,-bottom,-right)
```

```r
condition.subject.info <- all.data %>%
  filter(!is.na(condition)) %>%
  select(subject_id, condition) %>%
  group_by(subject_id) %>%
  mutate(block = c(1,2)) %>%
  ungroup()

task.data <- task.data %>%
  group_by(subject_id) %>%
  mutate(block = c(rep(1,n()/2),rep(2,n()/2))) %>%
  left_join(condition.subject.info, by=c("subject_id", "block"))
```

```r
free.view.data <- task.data %>%
  filter(condition == "free") %>%
  unnest(webgazer_data)
```

```r
free.view.data <- free.view.data %>%
  mutate(x.percent = x / width * 100, y.percent = y / height * 100) %>%
  mutate(view_quadrant = case_when(
    x.percent <= 50 & y.percent <= 50 ~ 'top.left',
    x.percent > 50 & y.percent <= 50 ~ 'top.right',
    x.percent <= 50 & y.percent > 50 ~ 'bottom.left',
```

```
    x.percent > 50 & y.percent > 50 ~ 'bottom.right'
  )) %>%
  mutate(normalized_quadrant = case_when(
    quadrant == 1 & view_quadrant == 'top.left' ~ 'critical',
    quadrant == 1 & view_quadrant == 'top.right' ~ 'first',
    quadrant == 1 & view_quadrant == 'bottom.right' ~ 'second',
    quadrant == 1 & view_quadrant == 'bottom.left' ~ 'third',

    quadrant == 2 & view_quadrant == 'top.left' ~ 'third',
    quadrant == 2 & view_quadrant == 'top.right' ~ 'critical',
    quadrant == 2 & view_quadrant == 'bottom.right' ~ 'first',
    quadrant == 2 & view_quadrant == 'bottom.left' ~ 'second',

    quadrant == 3 & view_quadrant == 'top.left' ~ 'first',
    quadrant == 3 & view_quadrant == 'top.right' ~ 'second',
    quadrant == 3 & view_quadrant == 'bottom.right' ~ 'third',
    quadrant == 3 & view_quadrant == 'bottom.left' ~ 'critical',

    quadrant == 4 & view_quadrant == 'top.left' ~ 'second',
    quadrant == 4 & view_quadrant == 'top.right' ~ 'third',
    quadrant == 4 & view_quadrant == 'bottom.right' ~ 'critical',
    quadrant == 4 & view_quadrant == 'bottom.left' ~ 'first'
  ))
```

Replication

```
free.view.summary.trial.data <- free.view.data %>%
  group_by(subject_id, trial_index, relation) %>%
  summarize(critical = sum(normalized_quadrant == 'critical')/n(),
            first = sum(normalized_quadrant == 'first')/n(),
            second = sum(normalized_quadrant == 'second')/n(),
            third = sum(normalized_quadrant == 'third')/n()) %>%
  pivot_longer(c("critical", "first", "second", "third"), names_to = "normalized_quadrant", values_to =
```

```
## `summarise()` has grouped output by 'subject_id', 'trial_index'. You can override using the `.groups
```

```
free.view.summary.subject.data <- free.view.summary.trial.data %>%
  group_by(subject_id,relation,normalized_quadrant) %>%
  summarize(proportion = mean(proportion))
```

```
## `summarise()` has grouped output by 'subject_id', 'relation'. You can override using the `.groups` a
```

Eye-gaze. Looks during the retrieval period were categorized as belonging to one of four quadrants based on the x,y coordinates. The critical quadrant was the one in which the to-be-retrieved object had been previously located during encoding. The other three quadrants were semi-randomly labeled “first”, “second,” third” (e.g., when the critical quadrant was in the top left, the “first” quadrant was the top right quadrant, but when the critical quadrant was in the top right, “first” corresponded to bottom right, etc.). In the free-viewing condition, participants directed a larger proportion of looks to the critical quadrant (see Figure @ref(fig:E2-gaze-fig)).

```
free.view.summary.condition1.data <- free.view.summary.subject.data %>%
  group_by(normalized_quadrant) %>%
  summarize(M = mean(proportion), SE = sd(proportion) / sqrt(n()))
```

```
fig<-ggplot(free.view.summary.condition1.data, aes(x=normalized_quadrant, y=M, ymax=M+SE, ymin=M-SE, fil
  geom_col(position=position_dodge(), color = "black")+
  geom_errorbar(position=position_dodge(width=0.9), width=0.1)+
  scale_fill_brewer(palette="Set1",)+
  theme_classic()+
  labs(x = "Quadrant", y = "Mean proportion of looks")+
  theme(legend.position = "none")
fig
```

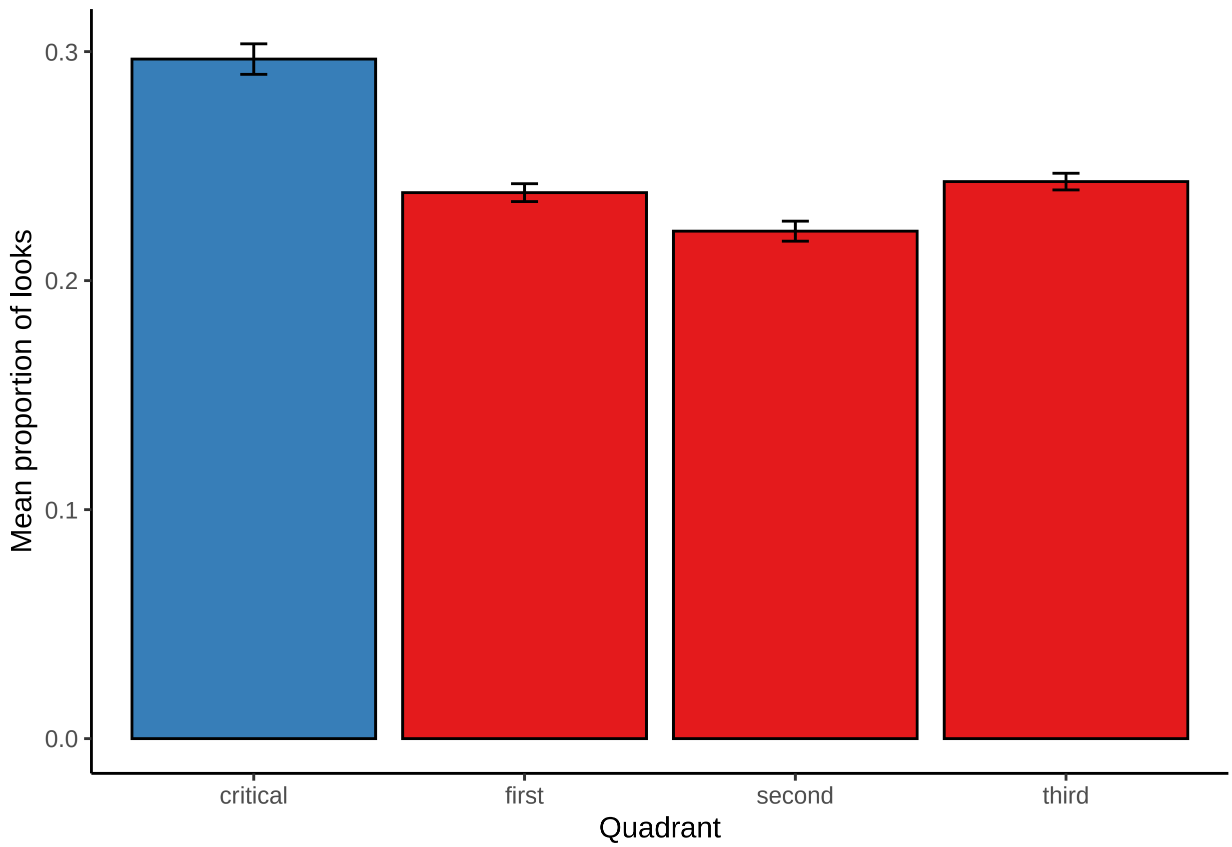

Figure 1: Proportion of eye-gaze to critical quadrant and other three quadrants durign memory retrieval.

```
saveRDS(fig, "output/ETfig.rds")
```

```
E2_gaze_model<-lmer(proportion ~ normalized_quadrant + (1+normalized_quadrant|subject_id), data = free.
#summary(m1)
E2_gaze_model_tab = broom.mixed::tidy(E2_gaze_model)
E2_gaze_model_q1 = E2_gaze_model_tab %>% filter(term == "normalized_quadrantfirst")
E2_gaze_model_q2 = E2_gaze_model_tab %>% filter(term == "normalized_quadrantsecond")
E2_gaze_model_q3 = E2_gaze_model_tab %>% filter(term == "normalized_quadrantthird")
```

The proportion of looks across quadrant was analyzed in linear mixed-effects model with quadrant as the predictor (critical as the reference level). The model included random intercepts and slopes for participants[1] Proportions of looks were significantly higher for the critical quadrant compared to the other three (first: b = -0.06, SE = 0.01, p<0.001, second: b = -0.08, SE = 0.01, p<0.001, third: b = -0.05, SE = 0.01, p<0.001)

```
behavioral.data <- task.data %>%
  select(subject_id, trial_index, relation, rt, response, correct_response, condition) %>%
  mutate(correct = response == correct_response)
```

```
# *note that paper computer accuracy as hit rate - false alarm rate*
acc.behavioral.subject.data <- behavioral.data %>%
  group_by(subject_id, relation, condition) %>%
  summarize(hit.rate = sum(correct == TRUE & response == 't') / sum(correct_response == 't'),
            fa.rate = sum(correct == FALSE & response == 't') / sum(correct_response == 'f')) %>%
  mutate(accuracy = hit.rate - fa.rate)
```

```
## `summarise()` has grouped output by 'subject_id', 'relation'. You can override using the `.groups` a
```

```
rt.behavioral.subject.data <- behavioral.data %>%
  group_by(subject_id, relation, condition) %>%
  filter(correct == TRUE) %>%
  summarize(rt = mean(rt))
```

```
## `summarise()` has grouped output by 'subject_id', 'relation'. You can override using the `.groups` a
```

```
acc.summary.condition.data <- acc.behavioral.subject.data %>%
  group_by(relation, condition) %>%
  summarize(M=mean(accuracy), SE=sd(accuracy)/sqrt(n()))
```

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

```
acc_fig<-ggplot(acc.summary.condition.data %>%
                 mutate(relation = factor(relation, levels = c("intra", "inter"))),
                        aes(x=relation, color=condition, y=M, ymax=M+SE, ymin=M-SE, group=condition))+
  geom_point(size=5)+
  geom_line()+
  geom_errorbar(width=0.2)+
  scale_color_manual(values=c("orange2","royalblue4"))+
  coord_cartesian(ylim=c(0.3,0.8)) +
  labs(x="Statement Type", y="Accuracy", color=NULL)+
  theme_classic()
```

```
rt.summary.condition.data <- rt.behavioral.subject.data %>%
  group_by(relation, condition) %>%
  summarize(M=mean(rt), SE=sd(rt)/sqrt(n()))
```

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

[1] `lme4` syntax: `lmer(proportion ~ quadrant + (1+quadrant|subject_id))`. Among other limitations, this approach may violate the independence assumptions of the linear model because looks to the four locations are not independent. This analysis was chosen because it is analogous to the ANOVA analysis conducted in the original paper.

```
rt_fig<-ggplot(rt.summary.condition.data %>%
                 mutate(relation = factor(relation, levels = c("intra", "inter"))),
               aes(x=relation, color=condition, y=M, ymax=M+SE, ymin=M-SE, group=condition))+
  geom_point(size=5)+
  geom_line()+
  geom_errorbar(width=0.2)+
  scale_color_manual(values=c("orange2","royalblue4"))+
  labs(x="Statement Type", y="Response Time (ms)", color=NULL)+
  theme_classic()
```

```
acc_rt_fig<-acc_fig + rt_fig + plot_layout(guides = "collect")
acc_rt_fig
```

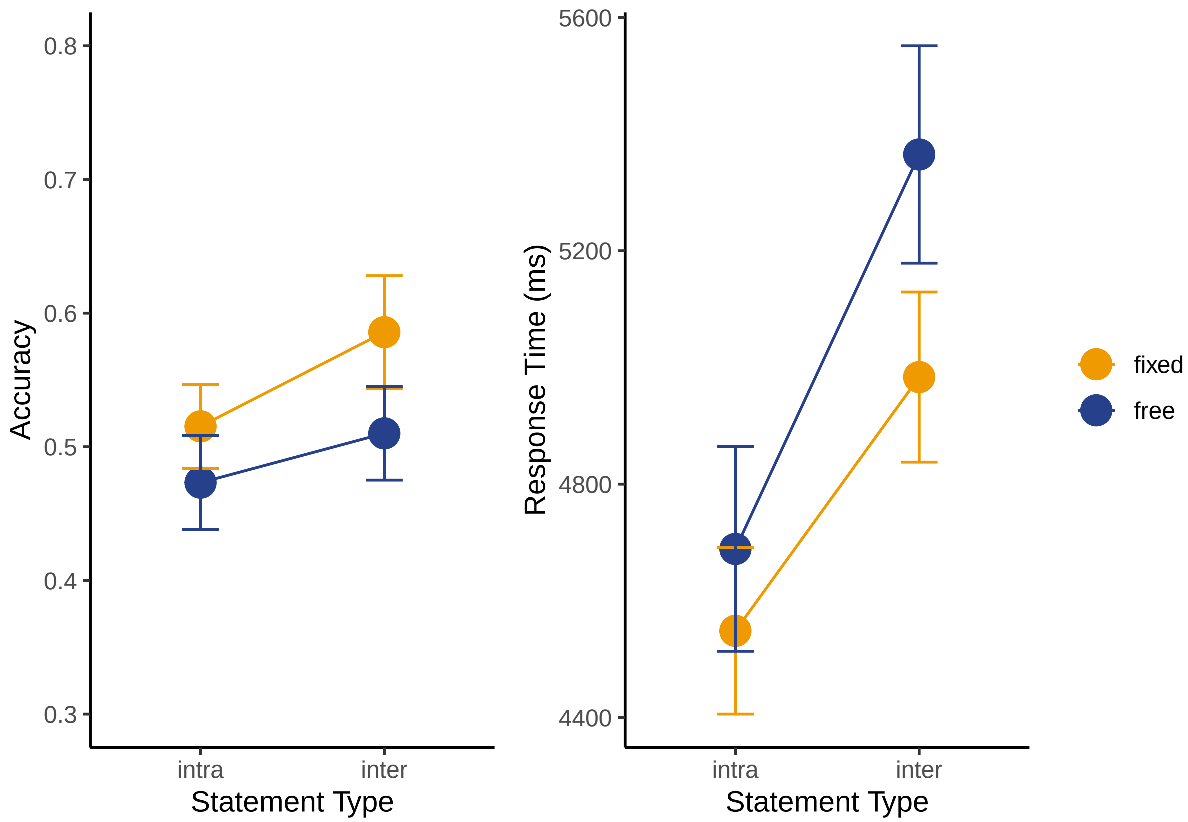

Figure 2: Accuracy and response times during memory retrieval.

```
acc.behavioral.subject.data$relation = factor(acc.behavioral.subject.data$relation)
contrasts(acc.behavioral.subject.data$relation)<-c(-0.5,0.5)
#contrasts(acc.behavioral.subject.data$relation)

acc.behavioral.subject.data$condition = factor(acc.behavioral.subject.data$condition)
contrasts(acc.behavioral.subject.data$condition)<-c(-0.5,0.5)
#contrasts(acc.behavioral.subject.data$condition)

E2_acc_model<-lmer(accuracy ~ relation*condition + (1|subject_id), data = acc.behavioral.subject.data)
```

```
#summary(m1)
E2_acc_model_tab = broom.mixed::tidy(E2_acc_model)
E2_acc_model_rel = E2_acc_model_tab %>% filter(term == "relation1")
E2_acc_model_cond = E2_acc_model_tab %>% filter(term == "condition1")
```

```
rt.behavioral.subject.data$relation = factor(rt.behavioral.subject.data$relation)
contrasts(rt.behavioral.subject.data$relation)<-c(-0.5,0.5)
#contrasts(rt.behavioral.subject.data$relation)

rt.behavioral.subject.data$condition = factor(rt.behavioral.subject.data$condition)
contrasts(rt.behavioral.subject.data$condition)<-c(-0.5,0.5)
#contrasts(rt.behavioral.subject.data$condition)

E2_RT_model<-lmer(rt ~ relation*condition + (1|subject_id), data = rt.behavioral.subject.data)
#summary(m1)
E2_RT_model_tab = broom.mixed::tidy(E2_RT_model)
E2_RT_model_rel = E2_RT_model_tab %>% filter(term == "relation1")
E2_RT_model_cond = E2_RT_model_tab %>% filter(term == "condition")
```

Response Time and Accuracy. Participants' response times and accuracies on memory questions are summarized in Figure @ref(fig:E2-rt-acc-fig). Both dependent variables were analyzed with linear mixed-effects model with relation type (interobject = -0.5, intraobject=0.5) and viewing_condition (fixed = -0.5, free=0.5) and their interaction as the predictors. The model included random intercepts for participants[2]. Response times were slower for interobject (e.g., "The train is to the right of the taxi.") than intraobject (e.g., "The train is facing right.") questions (b = -555.6, SE = 105.24, p<0.001). Response times were slower in the free viewing condition than the fixed condition (b = , SE = , p<0.001). Accuracy did not differ significantly between interobject and intraobject questions (b = -0.05, SE = 0.03, p=0.05). Participants were less accurate in the free viewing condition than the fixed condition (b = -0.06, SE = 0.03, p<0.03). The interaction ws not a signficant predictor for response times or accuracy.

```
#Goal: ANOVA with RT as DV, condition, statement as IVs
rt.anova <- ezANOVA(rt.behavioral.subject.data, dv=rt, wid=subject_id, within = c(relation, condition))
rt.anova$ANOVA
```

```
acc.anova <- ezANOVA(acc.behavioral.subject.data, dv=accuracy, wid=subject_id, within = c(relation, cond
acc.anova$ANOVA
```

Q: Can we recover item IDs to do crossed random effects

[2] `lme4` syntax: `lmer(DV ~ relation_type*viewing_condition + (1|subject_id))`