

Visual Search Familiar Faces: Figures

Matteo Visconti di Oleggio Castello

Figures

This R markdown file will produce the figures for the manuscript.

Load some useful libraries and load the data.

```
require(dplyr)
require(ggplot2)
require(broom)
require(doParallel)
require(foreach)
require(assertthat)

data <- read.csv('data.csv')
data_correct <- data %>% filter(correct == 1)
```

Set up some variables that will be used later.

```
nproc <- 4 # change this to use more/less processors for parallel use
seed <- 42 # seed for rng to obtain reproducible results in different runs
alpha <- .05 # significance level for confidence intervals
nbs <- 10000 # number of bootstrapping repetitions
```

Bootstrapping of the data

Bootstrap data resampling within each condition. It could take a while depending on the number of processors used.

```
cl <- makeCluster(nproc)
registerDoParallel(cl)
# push required packages to each worker
clusterCall(cl, function() library(magrittr))
clusterCall(cl, function() library(plyr))
clusterCall(cl, function() library(dplyr))
bstrap <- data.frame()
set.seed(seed)
bstrap <- foreach(i = 1:nbs, .combine=rbind) %dopar% {
  tmp <-
    data_correct %>%
    group_by(orientation, target_presence, familiarity,
              set_size, target_sex, subid) %>%
    sample_frac(1, replace=T) %>%
    group_by(orientation, target_presence, familiarity, set_size) %>%
    summarise(avg=mean(RT))

  tmp$index <- i
  tmp
}
```

```
}
stopCluster(cl)
```

Plot of Average Reaction Times

Now compute confidence intervals and averages from the bootstrapped samples.

```
cis <-
  bstrap %>%
  group_by(orientation, target_presence, familiarity, set_size) %>%
  summarise(low=quantile(avg, alpha/2), high=quantile(avg, 1-alpha/2))

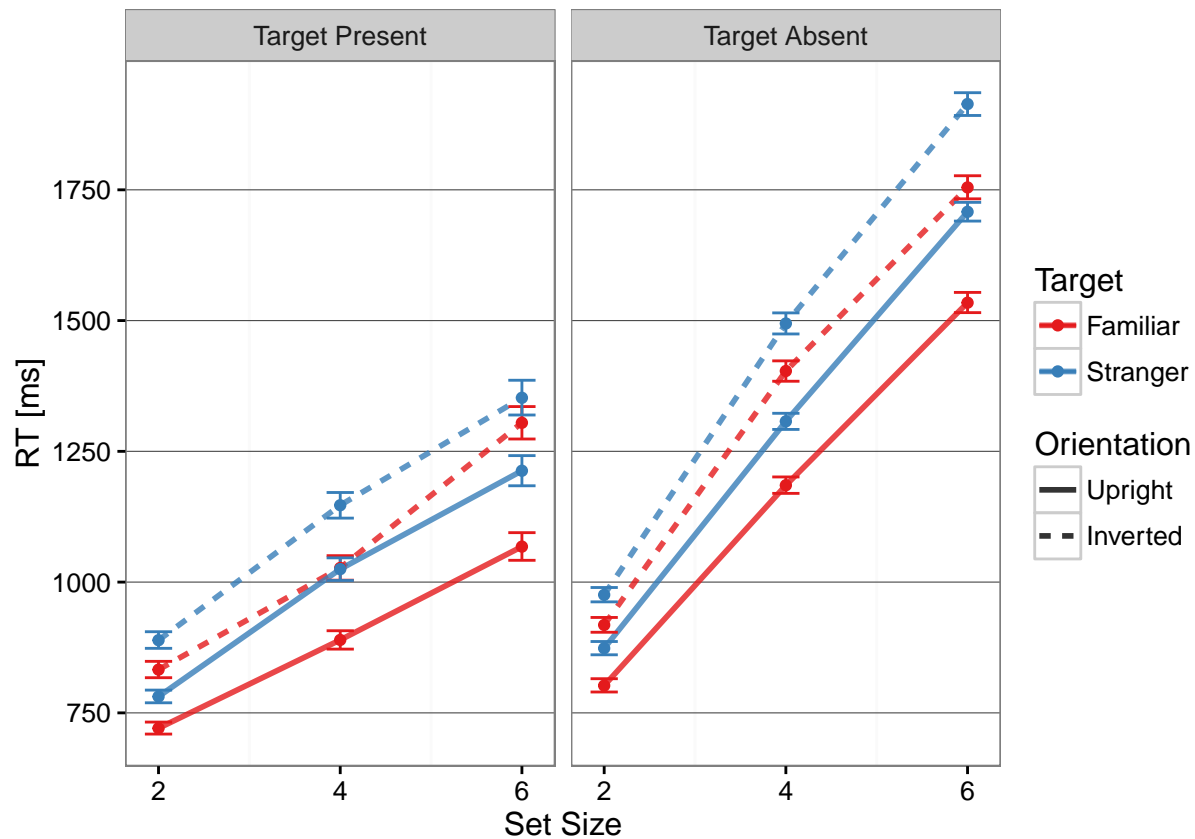
# compute average from original data
avgs <- data_correct %>%
  group_by(orientation, target_presence, familiarity, set_size) %>%
  summarise(avg=mean(RT))

cis <- merge(cis, avgs)
```

Now plot the figure

```
# first reorder the levels of target_presence and orientation
cis$orientation <- factor(cis$orientation, levels=c('Upright', 'Inverted'))
cis$target_presence <- factor(cis$target_presence,
                             levels=c('Target Present', 'Target Absent'))

ggplot(cis, aes(set_size, avg, ymin=low, ymax=high,
                 color=familiarity, linetype=orientation)) +
  geom_line(alpha=.8, size=1) +
  geom_errorbar(width=0.3, linetype='solid') +
  geom_point() +
  facet_grid(~ target_presence) +
  theme_bw(base_size=12) +
  theme(panel.grid.major.y = element_line(colour = "gray30"),
        panel.grid.major.x = element_blank(),
        panel.grid.minor.y = element_blank()) +
  scale_y_continuous(breaks=c(750, 1000, 1250, 1500, 1750)) +
  scale_x_continuous(breaks=c(2, 4, 6)) +
  labs(x='Set Size', y='RT [ms]', color='Target', linetype='Orientation') +
  scale_color_brewer(palette='Set1')
```



Plot of Effect Sizes

Compute difference Stranger - Familiar for each bootstrapped sample and obtain their confidence intervals.

```
bstrap_fam <- filter(bstrap, familiarity == 'Familiar') %>%
  select(-familiarity)
bstrap_str <- filter(bstrap, familiarity == 'Stranger') %>%
  select(-familiarity)
# assert that the order is the same
tocheck <- c('orientation', 'target_presence', 'set_size', 'index')
for (check in tocheck) {
  assert_that(all(bstrap_fam[, check] == bstrap_str[, check]))
}

bstrap_es <- bstrap_str
bstrap_es$avg <- bstrap_es$avg - bstrap_fam$avg

# compute cis for effect size
cis_es <-
  bstrap_es %>%
  group_by(orientation, target_presence, familiarity, set_size) %>%
  summarise(low=quantile(avg, alpha/2), high=quantile(avg, 1-alpha/2))
```

Now we do the same in the original data.

```

avg_fam <- avgs %>% filter(familiarity == 'Familiar') %>% select(-familiarity)
avg_str <- avgs %>% filter(familiarity == 'Stranger') %>% select(-familiarity)
# assert that the order is the same
tocheck <- c('orientation', 'target_presence', 'set_size')
for (check in tocheck) {
  assert_that(all(avg_fam[, check] == avg_str[, check]))
}

# get avg effect size
avg_es <- avg_str
avg_es$avg <- avg_es$avg - avg_fam$avg

```

Merge datasets and obtain data frame for plotting.

```

cis_es <- merge(cis_es, avg_es)
cis_es$orientation <- factor(cis_es$orientation,
                             levels=c('Upright', 'Inverted'))
cis_es$target_presence <- factor(cis_es$target_presence,
                                 levels=c('Target Present', 'Target Absent'))

```

Plot the data.

```

pd <- position_dodge(w=0.9)
ggplot(cis_es, aes(as.factor(set_size), avg, ymin=low, ymax=high,
                    fill=orientation, group=orientation)) +
  geom_bar(position=pd, stat='identity', color='black') +
  geom_errorbar(width=0.2, position=pd) +
  facet_grid(~target_presence) +
  scale_fill_brewer(palette='Set2') +
  theme_bw() +
  labs(x='Set Size', y='Stranger - Familiar Effect Size [ms]',
       fill='Orientation')

```

