Analysis of learning via the IAT using race stimuli II - Experiment 2

Ian Hussey

Contents

Conceptual replication of Experiment 1. Employs AMP as DV rather than SC-IAT.

Hypotheses

H1: Completing an IAT serves to train attitudes as well as test them. Participants who complete a race IAT will demonstrate more negative implicit evaluations of the outgroup (black people) on the AMP than participants who completed a control (flowers-insects) IAT.

H2: Completing an IAT serves to train attitudes as well as test them. Participants who complete a race IAT will demonstrate more negative self-reported evaluations of the outgroup (black people) on the ratings than participants who completed a control (flowers-insects) IAT.

H3: H2 will also be applied to a combination analysis of all self report ratings collected in both Experiments 1 and 2.

```
# dependencies
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```r
library(psych)
```

```
##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(afex)
```

```
## Loading required package: lme4
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:tidyr':
##
##     expand
## Loading required package: reshape2
##
## Attaching package: 'reshape2'
## The following object is masked from 'package:tidyr':
##
##     smiths
## Loading required package: lsmeans
## Loading required package: estimability
## ************
## Welcome to afex. Important changes in the current version:
## - Functions for ANOVAs have been renamed to: aov_car(), aov_ez(), and aov_4().
## - ANOVA functions return an object of class 'afex_aov' as default, see: ?aov_car
## - 'afex_aov' objects can be passed to lsmeans for contrasts and follow-up tests.
## - Reset previous (faster) behavior via: afex_options(return_aov='nice')
## - Many more arguments can now be set globally via options, see: afex_options()
## ************
```

```r
library(effsize)
library(weights)  # for rd(), a round() alternative
```

```
## Loading required package: Hmisc
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:psych':
##
##     describe
## The following objects are masked from 'package:dplyr':
##
##     combine, src, summarize
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
## Loading required package: gdata
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
##
## Attaching package: 'gdata'
## The following object is masked from 'package:Hmisc':
##
##     combine
## The following objects are masked from 'package:dplyr':
##
##     combine, first, last
## The following object is masked from 'package:purrr':
##
##     keep
## The following object is masked from 'package:stats':
##
##     nobs
## The following object is masked from 'package:utils':
##
##     object.size
## The following object is masked from 'package:base':
##
##     startsWith
## Loading required package: mice
## Loading required package: Rcpp
## mice 2.25 2015-11-09
##
## Attaching package: 'mice'
## The following object is masked from 'package:tidyr':
##
##     complete
```

```
library(plotrix)  # for std.error
```

```
##
## Attaching package: 'plotrix'
```

```
## The following object is masked from 'package:psych':
##
##     rescale
```

```
library(lme4)
library(effects)
```

Descriptive statistics

All participants

Descriptive data for sample.

```
setwd(params$location_of_data)

data_df <-
  read.csv("processed data/wide data.csv") %>%
  mutate(gender = as.factor(gender))

colnames(data_df)
```

```
##  [1] "participant"                       "amp_recognition_response"
##  [3] "condition"                         "IAT_condition"
##  [5] "block_order"                       "task_order"
##  [7] "gender"                            "age"
##  [9] "modern_racism_scale_total"         "IAT_mean_RT"
## [11] "IAT_perc_acc"                      "IAT_exclude_based_on_fast_trials"
## [13] "AMP_mean_RT"                       "AMP_perc_acc"
## [15] "exclude"
```

```
data_df %>% dplyr::count(gender)
```

```
## # A tibble: 8 × 2
##       gender     n
##       <fctr> <int>
## 1                1
## 2        fee     1
## 3      female    99
## 4     female     2
## 5 genderfluid    1
## 6        male   108
## 7       male     1
## 8       woman    1
```

```
data_df %>%
  dplyr::select(age) %>%
  psych::describe(fast = TRUE,  # subset of descriptive stats
                  ranges = FALSE,
                  trim = 0) %>%
  dplyr::select(-vars, -se)
```

```
##       n  mean   sd
## age 214 35.75 12.1
```

Sample descriptive statistics

```
passers_df <-
  data_df %>%
  dplyr::filter(exclude == FALSE)

passers_df %>% dplyr::count(IAT_condition)
```

```
## # A tibble: 2 × 2
##          IAT_condition     n
##                 <fctr> <int>
## 1 Flowers-Insects IAT   106
## 2            Race IAT   107
```

```
passers_df %>%
  dplyr::select(IAT_mean_RT,
                IAT_perc_acc,
                AMP_mean_RT,
                AMP_perc_acc) %>%
  psych::describe(fast = TRUE,  # subset of descriptive stats
                  ranges = FALSE,
                  trim = 0) %>%
  dplyr::select(-vars, -se)
```

```
##                n   mean     sd
## IAT_mean_RT  213 837.23 158.79
## IAT_perc_acc 213   0.93   0.06
## AMP_mean_RT  213 543.05 194.24
## AMP_perc_acc 213   0.65   0.21
```

H1: Differences in IAT effects between contrast pair conditions

```
# get data
setwd(params$location_of_data)

AMP_data <-
  read.csv("processed data/long AMP data.csv") %>%
  dplyr::filter(exclude == FALSE) %>%
  dplyr::mutate(participant = as.factor(participant),
                rating_factor = as.factor(ifelse(accuracy == 0, "negative",
                                                 ifelse(accuracy == 1, "positive", NA)))) %>%
  dplyr::rename(prime_type = trialcode,
                rating = accuracy)
```

Descriptive statistics

```
AMP_summary_data <-
  AMP_data %>%
  group_by(IAT_condition, prime_type) %>%
  dplyr::summarize(mean_rating = round(mean(rating), 2),
                   sd_rating = round(sd(rating), 2),
                   se_rating = round(std.error(rating), 2))
```

```
AMP_summary_data
```

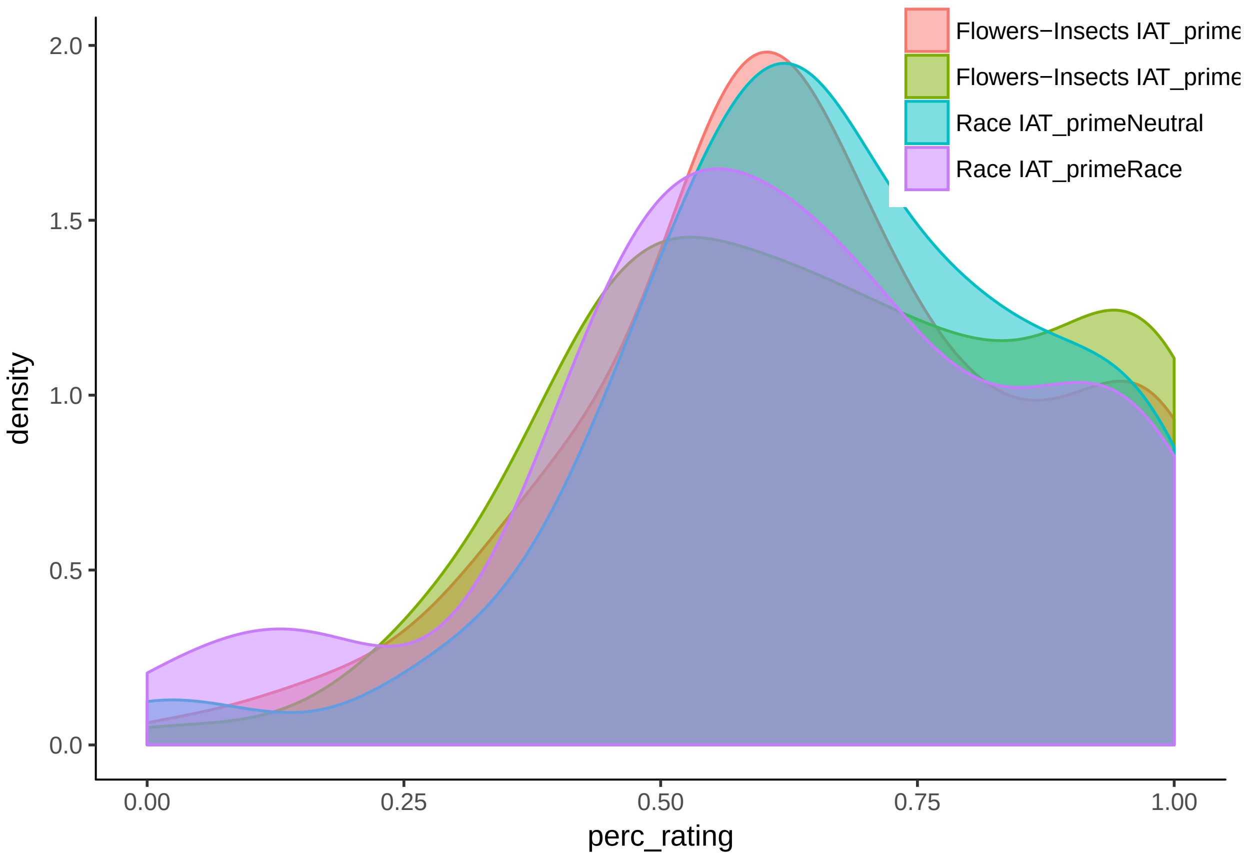
```
## Source: local data frame [4 x 5]
## Groups: IAT_condition [?]
##
##         IAT_condition   prime_type mean_rating sd_rating se_rating
##                <fctr>       <fctr>       <dbl>     <dbl>     <dbl>
## 1 Flowers-Insects IAT primeNeutral        0.65      0.48      0.01
## 2 Flowers-Insects IAT    primeRace        0.67      0.47      0.01
## 3            Race IAT primeNeutral        0.66      0.47      0.01
## 4            Race IAT    primeRace        0.62      0.49      0.01
```

Plots

```
# apa theme for all plots
apatheme <-
  theme_bw() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        panel.border = element_blank(),
        #text = element_text(family='Arial'),  # doesn't play nice with knittr
        legend.title = element_blank(),
        legend.position = c(.9,.9),
        axis.line.x = element_line(color='black'),
        axis.line.y = element_line(color='black'))

# reshape and add a combined condition*IAT block variable for plotting
AMP_participant_summary_data <-
  AMP_data %>%
  group_by(participant, IAT_condition, prime_type) %>%
  dplyr::summarize(perc_rating = round(mean(rating), 2)) %>%
  ungroup() %>%
  group_by(IAT_condition, prime_type) %>%
  dplyr::mutate(mean_rating = round(mean(perc_rating), 2),
                sd_rating = round(sd(perc_rating), 2),
                se_rating = round(std.error(perc_rating), 2)) %>%
  ungroup() %>%
  dplyr::mutate(exp_factor = paste(IAT_condition, prime_type, sep = "_"))
```

Density plot split by factor

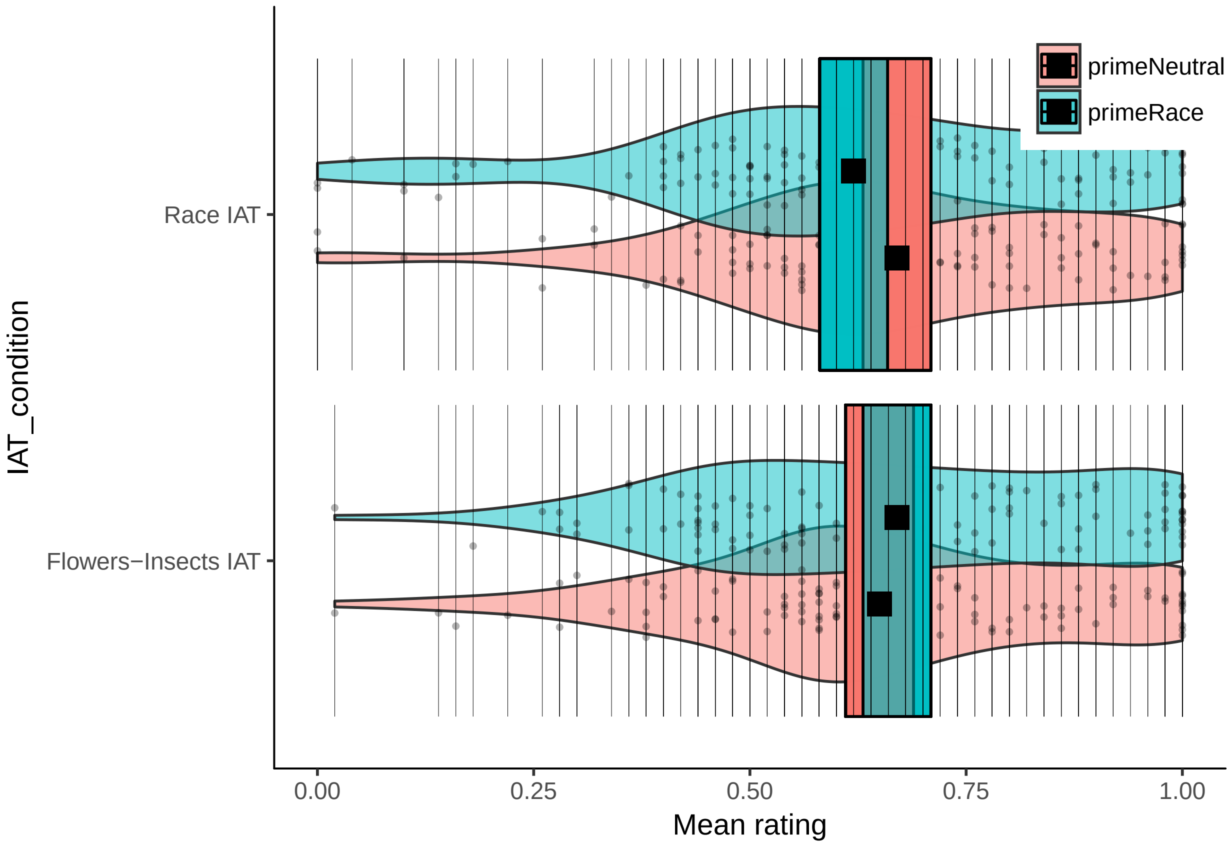
```
ggplot(AMP_participant_summary_data,
       aes(perc_rating, colour = exp_factor, fill = exp_factor)) +
  geom_density(alpha=0.50) +
  apatheme
```



Distribution and inference plot

Black squares are means, horizontal lines are 95% CIs, coloured shapes are distributions. I've chosen to omit presenting jittered raw data as it looks overplotted. NB scale was limited to 250 to 1250ms to make it more informative, although (non-outlier) values extend beyond the visible plot (and are included in the analysis).

```
ggplot(data = AMP_participant_summary_data,
       aes(x = IAT_condition, y = perc_rating, fill = prime_type)) +
  geom_violin(alpha = 0.5,
              position = position_dodge(width = .5)) +
  geom_point(size = 1,
             shape = 16,
             alpha = 0.3,
             position = position_jitterdodge(dodge.width = .5)) +
  geom_crossbar(aes(ymax = mean_rating + (1.96*se_rating),
                    ymin = mean_rating + (-1.96*se_rating)),
                alpha = 0.5,
                fatten = 0) +
  geom_point(aes(y = mean_rating),
             size = 4,
             shape = 15,
             position = position_dodge(width = .5)) +
  apatheme +
  ylab("Mean rating") +
  #coord_cartesian(ylim = c(250,1250))
  coord_flip()
```



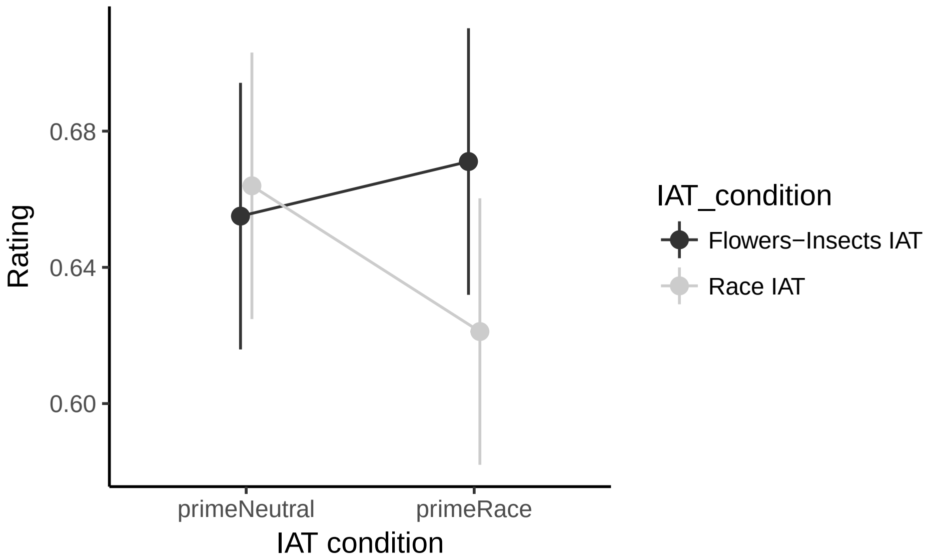
Greyscale inference plot - marginal means

```
# calculate marginal means
model_1_forplot <- lmer(rating ~ prime_type * IAT_condition + modern_racism_scale_total + (1 | participa
                        contrasts = list(prime_type = "contr.sum", IAT_condition = "contr.sum"),
                        data = AMP_data)

m1_marginal_means <- as.data.frame(effect("prime_type:IAT_condition", model_1_forplot))
m1_marginal_means
```

```
##     prime_type       IAT_condition       fit         se     lower
## 1 primeNeutral Flowers-Insects IAT 0.6550398 0.01997148 0.6158942
## 2    primeRace Flowers-Insects IAT 0.6710775 0.01997148 0.6319319
## 3 primeNeutral            Race IAT 0.6639469 0.01996345 0.6248170
## 4    primeRace            Race IAT 0.6211431 0.01996345 0.5820133
##       upper
## 1 0.6941854
## 2 0.7102231
## 3 0.7030767
## 4 0.6602730
```

```
ggplot(data = m1_marginal_means,
       aes(x = prime_type, y = fit, colour = IAT_condition)) +
  geom_pointrange(aes(ymax = upper,
                      ymin = lower),
                  position = position_dodge(width = .1)) +
  geom_line(aes(group = IAT_condition),
```

```
          position = position_dodge(width = .1)) +
  ylab("Rating") +
  xlab("IAT condition") +
  scale_colour_grey() +
  theme_classic()
```



Preregistered hypothesis test

Two important considerations:

1. Production of *p* values is contentious. The following suggests that parametric bootstrapping and the kenward rogers method give better error control than liklihood ratios, however KR and PB both throw errors on binomial data (i.e., AMP rating is binary). We therefore employ LR throughout for the sake of consistency. http://link.springer.com/article/10.3758%2Fs13428-016-0809-y

2. No effect sizes are produced due to contention over how to use the random factor error. See http://stats.stackexchange.com/questions/95054/how-to-get-an-overall-p-value-and-effect-size-for-a-categorical-factor-in-a-mi

The model is `rating_factor ~ prime_type * IAT_condition + modern_racism_scale_total + (1 | participant)`. That is, rating is predicted by the interaction between AMP prime type and training IAT condition, after controlling for differences in racism, and while allowing for participants to have a random intercept (i.e., acknowledging the non-independence of the mutliple ratings provided by each participant).

Our preregistered *a priori* hypothesis, that AMP effects will differ between conditions, relates to the interaction effect and not the main effects. We therefore employ type 3 sum of squares and examine only the results of the interaction.

```
# Check that variables that should be factors are indeed factors
sapply(AMP_data, class)
```

```
##                  participant                        rating
##                     "factor"                     "integer"
##                           rt                    prime_type
##                    "integer"                      "factor"
##                IAT_condition                   block_order
##                     "factor"                      "factor"
##                   task_order                        gender
##                     "factor"                      "factor"
```

```
##                              age        modern_racism_scale_total
##                        "integer"                        "integer"
##         amp_recognition_response IAT_exclude_based_on_fast_trials
##                         "factor"                        "logical"
##                          exclude                    rating_factor
##                        "logical"                         "factor"
```

```
# LME analysis
model_1 <- afex::mixed(rating ~ prime_type * IAT_condition + modern_racism_scale_total + (1 | participa
                       data = AMP_data,
                       family = binomial,
                       method = "LR")
```

```
## Fitting 5 (g)lmer() models:
## [.....]
```

```
model_1$anova_table
```

```
## Mixed Model Anova Table (Type 3 tests)
##
## Model: rating ~ prime_type * IAT_condition + modern_racism_scale_total +
## Model:     (1 | participant)
## Data: AMP_data
## Df full model: 6
##                           Df   Chisq Chi Df Pr(>Chisq)
## prime_type                 5  4.7688      1    0.02898 *
## IAT_condition              5  1.1507      1    0.28339
## modern_racism_scale_total  5  2.6265      1    0.10509
## prime_type:IAT_condition   5 24.4294      1  7.708e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# sigma/z scores
H1_z_score <- qnorm(1-model_1$anova_table$`Pr(>Chisq)`[4]) # 4th member is interaction effect
H1_z_score
```

```
## [1] 4.805762
```

H2: Differences in self-reported ratings between contrast pair conditions

Descriptive statistics

```
# get data
setwd(params$location_of_data)

ratings_data <-
  read.csv("processed data/long ratings data.csv") %>%
  dplyr::filter(exclude == FALSE) %>%
  dplyr::mutate(participant = as.factor(participant),
                IAT_condition = as.factor(IAT_condition))

ratings_summary_data <-
  ratings_data %>%
  group_by(IAT_condition) %>%
```

```r
  dplyr::summarize(mean_rating = round(mean(rating), 2),
                   sd_rating = round(sd(rating), 2),
                   se_rating = round(std.error(rating), 2))
```

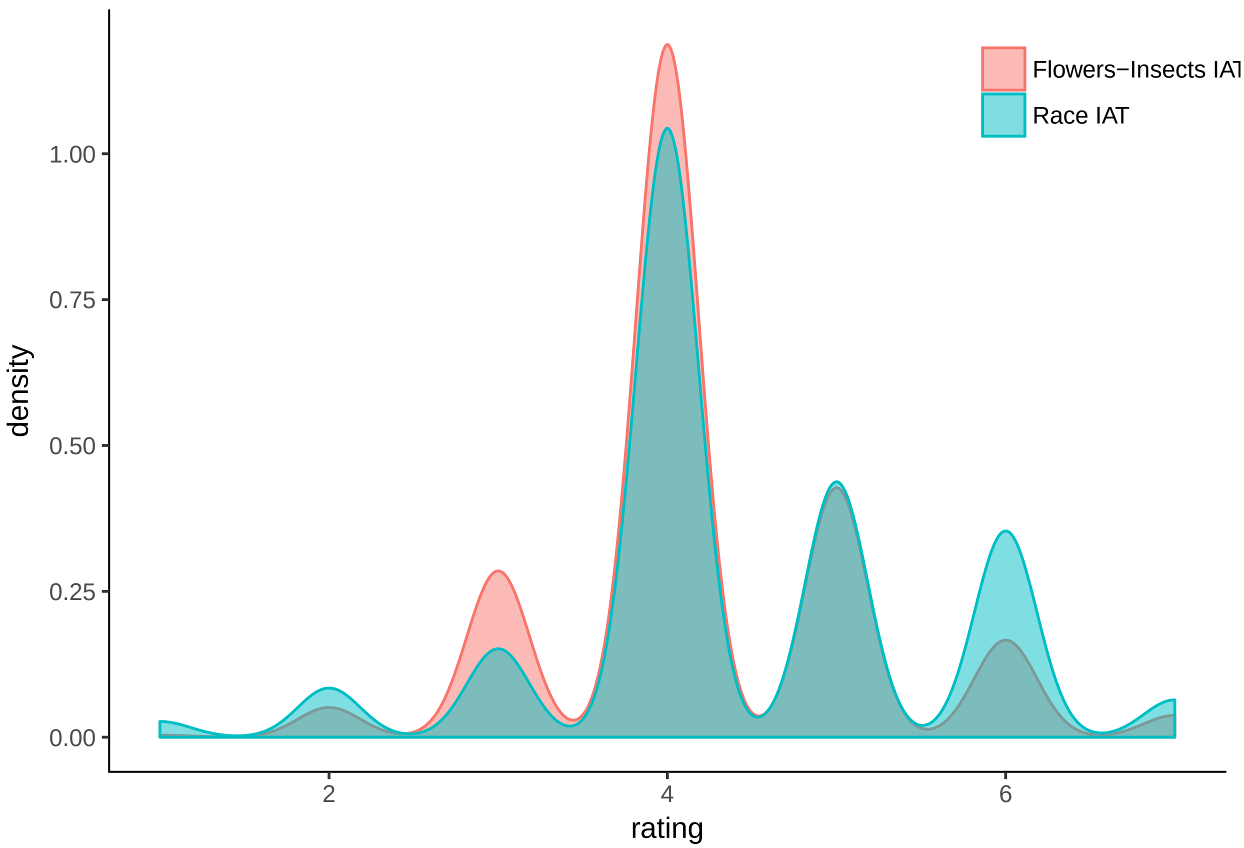
```r
ratings_summary_data
```

```
## # A tibble: 2 × 4
##          IAT_condition mean_rating sd_rating se_rating
##                 <fctr>       <dbl>     <dbl>     <dbl>
## 1 Flowers-Insects IAT        4.22      0.92      0.04
## 2            Race IAT        4.43      1.13      0.04
```

Plots

Density plot split by factor

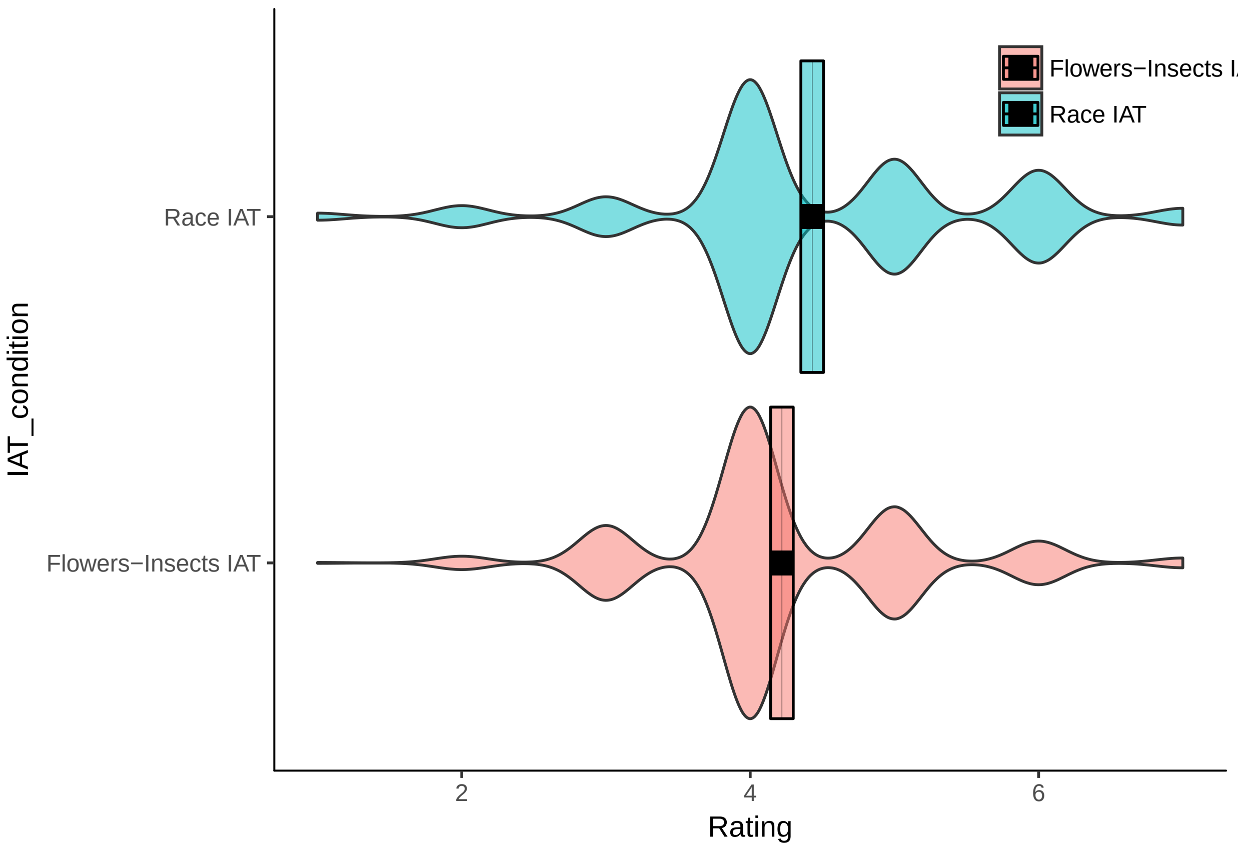
```r
ggplot(ratings_data,
       aes(rating, colour = IAT_condition, fill = IAT_condition)) +
  geom_density(alpha=0.50) +
  apatheme
```



Distribution and inference plot

Black squares are means, horizontal lines are 95% CIs, coloured shapes are distributions. I've chosen to omit presenting jittered raw data as it looks overplotted. NB scale was limited to 250 to 1250ms to make it more informative, although (non-outlier) values extend beyond the visible plot (and are included in the analysis).

```r
ggplot(data = ratings_summary_data,
       aes(x = IAT_condition, y = mean_rating, fill = IAT_condition)) +
  geom_violin(data = ratings_data,
              aes(x = IAT_condition, y = rating, fill = IAT_condition),
              alpha = 0.5,
              position = position_dodge(width = .5)) +
  geom_crossbar(aes(ymax = mean_rating + (1.96*se_rating),
                    ymin = mean_rating + (-1.96*se_rating)),
                alpha = 0.5,
                fatten = 0) +
  geom_point(size = 4,
             shape = 15,
             position = position_dodge(width = .5)) +
  apatheme +
  ylab("Rating") +
  #coord_cartesian(ylim = c(250,1250))
  coord_flip()
```



Greyscale inference plot - marginal means

```r
# calculate marginal means
model_2_forplot <- lmer(rating ~ IAT_condition + modern_racism_scale_total + (1 | participant),
                        contrasts = list(block = "contr.sum", IAT_condition = "contr.sum"),
                        data = ratings_data)

## Warning in model.matrix.default(fixedform, fr, contrasts): variable 'block'
## is absent, its contrast will be ignored
```

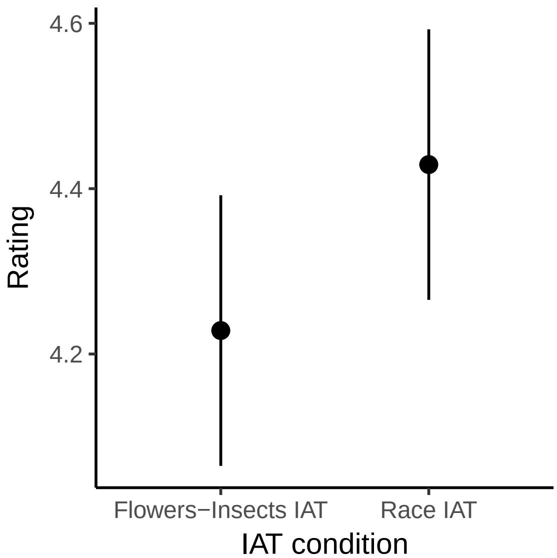
```r
m2_marginal_means <- as.data.frame(effect("IAT_condition", model_2_forplot))
```

```
## Warning in model.matrix.default(mt, mf, contrasts): variable 'block' is
## absent, its contrast will be ignored
```

```r
m2_marginal_means
```

```
##         IAT_condition      fit         se    lower    upper
## 1 Flowers-Insects IAT 4.228386 0.08344017 4.064691 4.392081
## 2            Race IAT 4.429147 0.08341480 4.265502 4.592792
```

```r
ggplot(data = m2_marginal_means,
       aes(x = IAT_condition, y = fit)) +
  geom_pointrange(aes(ymax = upper,
                      ymin = lower)) +
  ylab("Rating") +
  xlab("IAT condition") +
  scale_colour_grey() +
  theme_classic()
```



Preregistered hypothesis test

The model is `rating ~ IAT_condition + (1 | participant)`. That is, rating is predicted by training IAT condition, after controlling for racism and while allowing participants to have a random intercept (i.e., acknowledgin the non-independence of participants ratings of the images).

Our preregistered *a priori* hypothesis, that ratings would differ between conditions, relates to the main effect for IAT_condition. We therefore examine only the results of this main effect.

```r
# Check that variables that should be factors are indeed factors
sapply(ratings_data, class)
```

```
##                  participant                            trial_n
##                     "factor"                          "integer"
##                       rating                      IAT_condition
##                    "integer"                           "factor"
##                  block_order                         task_order
##                     "factor"                           "factor"
```

```
##                          gender                                  age
##                        "factor"                            "integer"
##        modern_racism_scale_total        amp_recognition_response
##                       "integer"                             "factor"
## IAT_exclude_based_on_fast_trials                             exclude
##                       "logical"                            "logical"
```

```
# LME analysis
model_2 <- afex::mixed(rating ~ IAT_condition + modern_racism_scale_total + (1 | participant),
                       data = ratings_data,
                       method = "LR")
```

```
## Fitting 3 (g)lmer() models:
## [...]
```

```
model_2$anova_table
```

```
## Mixed Model Anova Table (Type 3 tests)
##
## Model: rating ~ IAT_condition + modern_racism_scale_total + (1 | participant)
## Data: ratings_data
## Df full model: 5
##                           Df  Chisq Chi Df Pr(>Chisq)
## IAT_condition              4 2.9125      1    0.08790 .
## modern_racism_scale_total  4 6.1668      1    0.01302 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# sigma/z scores
H2_z_score <- qnorm(1-model_2$anova_table$`Pr(>Chisq)`[1]) # 1st member is IAT condition
H2_z_score
```

```
## [1] 1.353828
```

H3: Differences in self-reported ratings between contrast pair conditions - combination analysis across Experiments 1 and 2

Descriptive statistics

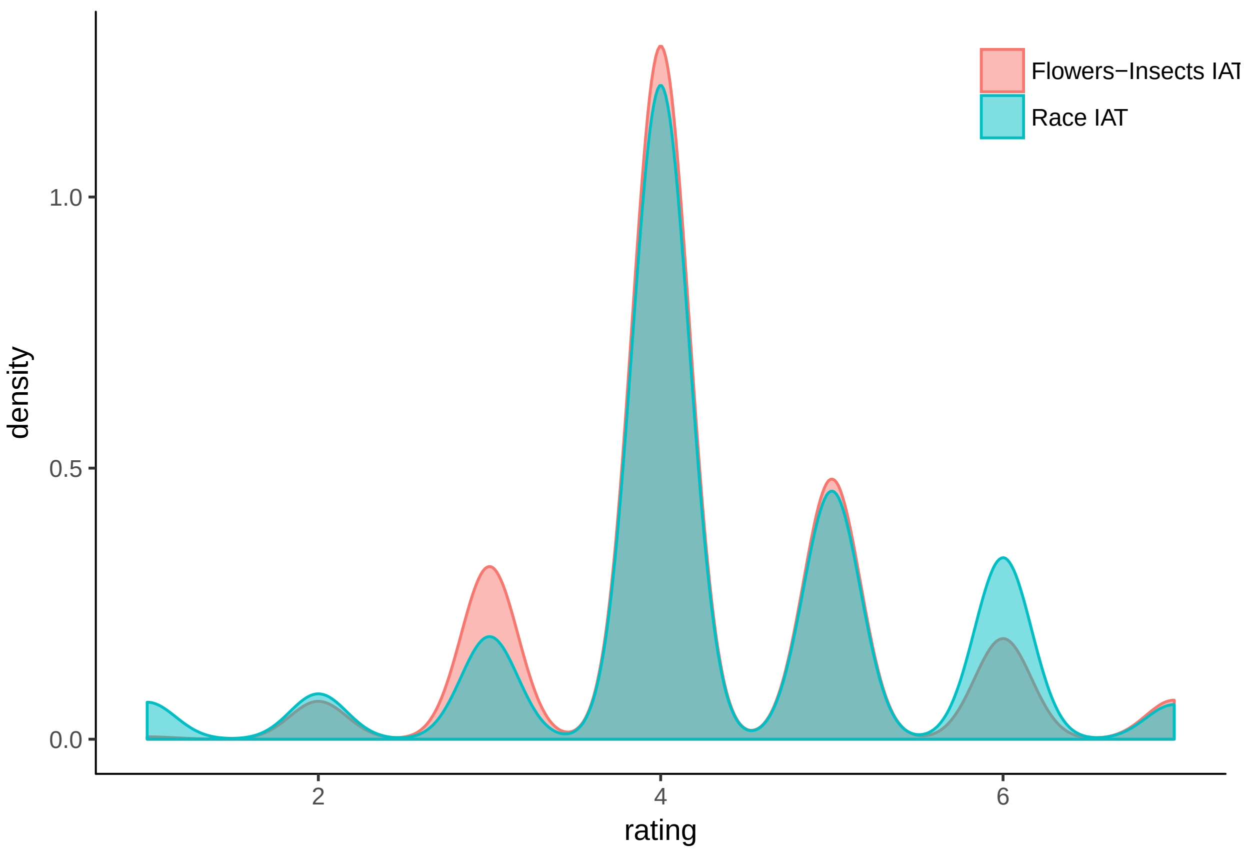
```
# trim exp 2 data
ratings_data_exp2 <-
  ratings_data %>%
  dplyr::mutate(experiment = 2,
                unique_id = as.factor(paste(experiment, participant, sep = "_"))) %>%
  dplyr::select(unique_id, experiment, IAT_condition, exclude, rating, modern_racism_scale_total)

# get data from exp 1
setwd(params$location_of_exp_1_data)

ratings_data_exp1 <-
  read.csv("processed data/long ratings data.csv") %>%
  #dplyr::filter(exclude == FALSE) %>%
  dplyr::mutate(experiment = 1,
                unique_id = as.factor(paste(experiment, participant, sep = "_")),
                IAT_condition = as.factor(IAT_condition),
```

```
                exclude = ifelse(IAT_exclude_based_on_fast_trials == TRUE, TRUE,
                                 ifelse(SCIAT_exclude_based_on_fast_trials == TRUE, TRUE,
                                        FALSE))) %>%
  dplyr::select(unique_id, experiment, IAT_condition, exclude, rating, modern_racism_scale_total)

combined_ratings_data <- rbind(ratings_data_exp1, ratings_data_exp2)


combined_ratings_summary_data <-
  combined_ratings_data %>%
  group_by(IAT_condition) %>%
  dplyr::summarize(mean_rating = round(mean(rating), 2),
                   sd_rating = round(sd(rating), 2),
                   se_rating = round(std.error(rating), 2))
```

Plots

Density plot split by factor

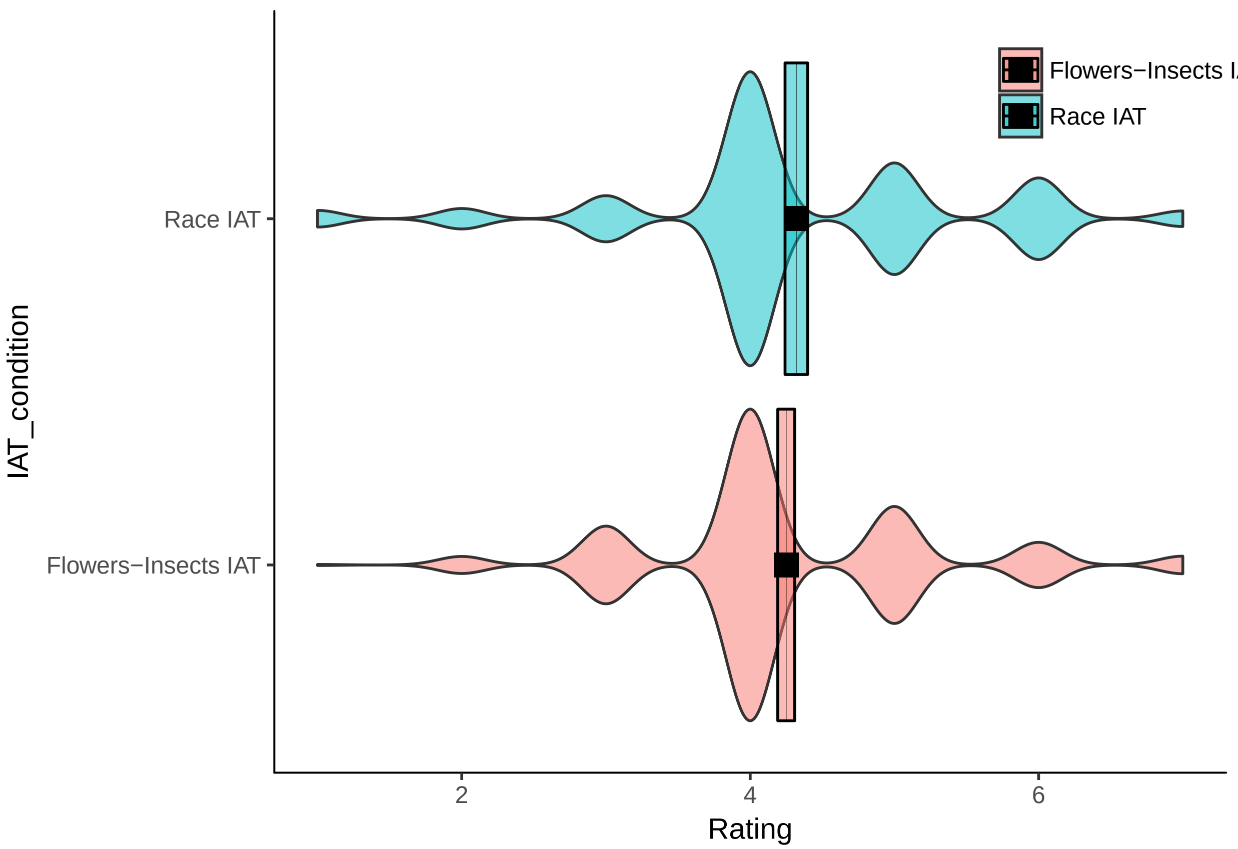
```
ggplot(combined_ratings_data,
       aes(rating, colour = IAT_condition, fill = IAT_condition)) +
  geom_density(alpha=0.50) +
  apatheme
```



Distribution and inference plot

Black squares are means, horizontal lines are 95% CIs, coloured shapes are distributions. I've chosen to omit presenting jittered raw data as it looks overplotted. NB scale was limited to 250 to 1250ms to make it more informative, although (non-outlier) values extend beyond the visible plot (and are included in the analysis).

```
ggplot(data = combined_ratings_summary_data,
       aes(x = IAT_condition, y = mean_rating, fill = IAT_condition)) +
  geom_violin(data = combined_ratings_data,
              aes(x = IAT_condition, y = rating, fill = IAT_condition),
              alpha = 0.5,
              position = position_dodge(width = .5)) +
  geom_crossbar(aes(ymax = mean_rating + (1.96*se_rating),
                    ymin = mean_rating + (-1.96*se_rating)),
                alpha = 0.5,
                fatten = 0) +
  geom_point(size = 4,
             shape = 15,
             position = position_dodge(width = .5)) +
  apatheme +
  ylab("Rating") +
  #coord_cartesian(ylim = c(250,1250))
  coord_flip()
```



Preregistered hypothesis test

The model is `rating ~ IAT_condition + (1 | participant)`. That is, rating is predicted by training IAT condition, after controlling for racism and while allowing participants to have a random intercept (i.e., acknowledgin the non-independence of participants ratings of the images).

Our preregistered *a priori* hypothesis, that ratings would differ between conditions, relates to the main effect for IAT_condition. We therefore examine only the results of this main effect.

```
# Check that variables that should be factors are indeed factors
sapply(ratings_data, class)
```

```
##                      participant                          trial_n
##                         "factor"                        "integer"
##                           rating                    IAT_condition
##                        "integer"                         "factor"
##                      block_order                       task_order
##                         "factor"                         "factor"
##                           gender                              age
##                         "factor"                        "integer"
##        modern_racism_scale_total         amp_recognition_response
##                        "integer"                         "factor"
## IAT_exclude_based_on_fast_trials                          exclude
##                        "logical"                        "logical"
```

```
# LME analysis
model_3 <- afex::mixed(rating ~ IAT_condition + modern_racism_scale_total + (1 | unique_id) + (1 | expe
                       data = combined_ratings_data,
                       method = "LR")
```

```
## Fitting 3 (g)lmer() models:
## [...]
```

```
model_3$anova_table
```

```
## Mixed Model Anova Table (Type 3 tests)
##
## Model: rating ~ IAT_condition + modern_racism_scale_total + (1 | unique_id) +
## Model:     (1 | experiment)
## Data: combined_ratings_data
## Df full model: 6
##                           Df   Chisq Chi Df Pr(>Chisq)
## IAT_condition              5  0.2279      1     0.6331
## modern_racism_scale_total  5 39.2949      1  3.644e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# sigma/z scores
H3_z_score <- qnorm(1-model_3$anova_table$`Pr(>Chisq)`[1]) # 1st member is IAT condition
H3_z_score
```

```
## [1] -0.3400176
```