# The IAT as an analogical learning task: Experiment 1

*Ian Hussey*

## Contents



Note that R treats the two conditions alphabetically (i.e., Insects, Flowers), so that all effects sizes are retured as negative despite being in line with the hypotheses. All are inverted when reported in the manuscript to make the reported results congruent with the wording of the hypothesis.

```
## Dependencies
library(tidyverse)
library(psych)
library(effsize)
library(lsr)  # for eta sq
library(MBESS)  # for ci.pvaf(), 95% CI on eta2
library(BayesFactor)

## Data acquisition
data_df <- read.csv("/Users/Ian/Dropbox/Work/Manuscripts/Hussey & De Houwer - the IAT as an analogical l
```

## Descriptive statistics

Gender counts

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

```
## # A tibble: 2 × 2
##   gender     n
##   <fctr> <int>
## 1      f    37
## 2      m    15
```

Descriptives for all participants

```
data_df %>%
  select(age,
         IAT_accuracy,
         IAT_mean_RT) %>%
  psych::describe(fast = TRUE,  # subset of descriptive stats
```

```
                 ranges = FALSE,
                 trim = 0)
```

```
##                vars  n   mean    sd    se
## age               1 52  22.06  3.46  0.48
## IAT_accuracy      2 52  92.80  4.62  0.64
## IAT_mean_RT       3 52 659.23 92.52 12.83
```

Descriptives by experimental condition

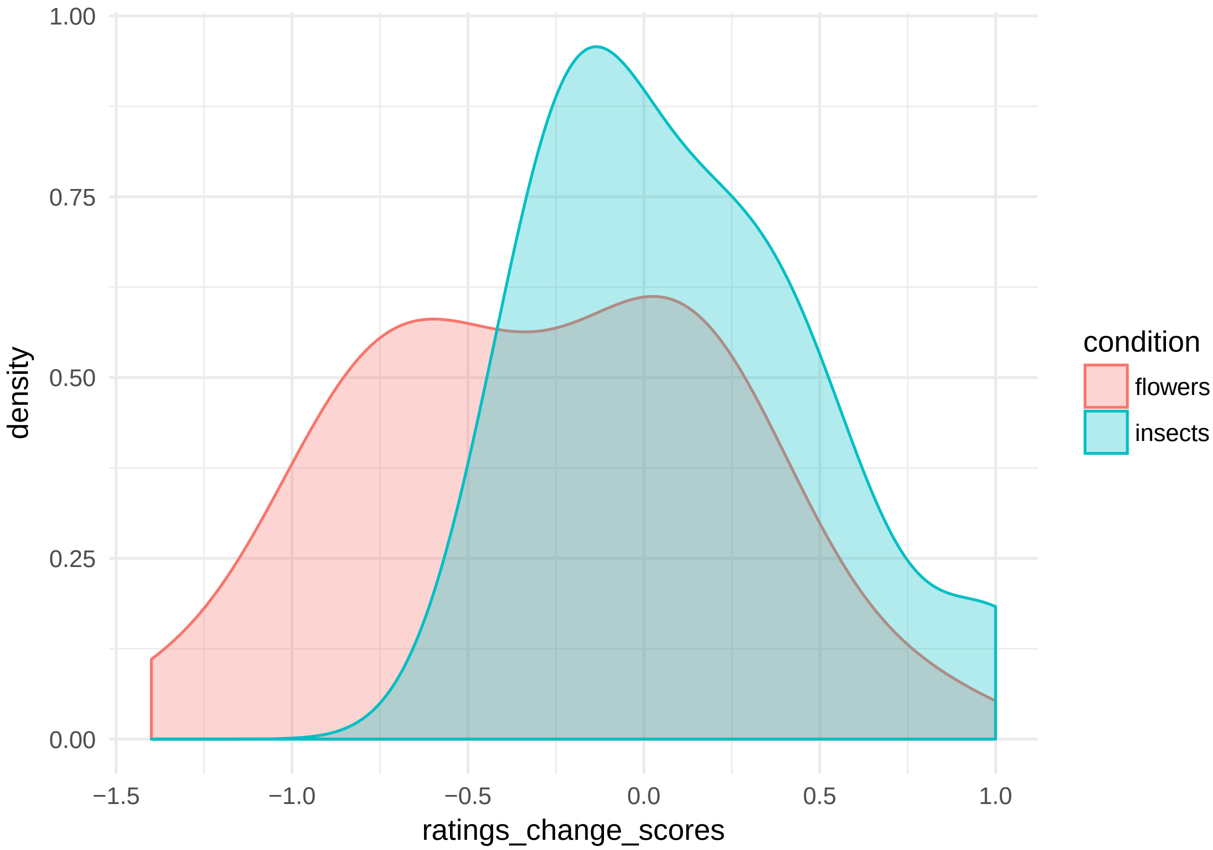
```
data_df %>%
  select(gender,
         age,
         ratings_pre,
         ratings_post,
         ratings_change_scores,
         D1) %>%
  psych::describeBy(data_df$condition,
                   fast = TRUE,  # subset of descriptive stats
                   ranges = FALSE,
                   trim = 0)
```

```
## $flowers
##                         vars  n  mean   sd   se
## gender*                    1 26   NaN   NA   NA
## age                        2 26 22.00 2.56 0.50
## ratings_pre                3 26  2.85 0.52 0.10
## ratings_post               4 26  2.55 0.63 0.12
## ratings_change_scores      5 26 -0.29 0.53 0.10
## D1                         6 26 -0.33 0.41 0.08
##
## $insects
##                         vars  n  mean   sd   se
## gender*                    1 26   NaN   NA   NA
## age                        2 26 22.12 4.22 0.83
## ratings_pre                3 26  2.92 0.42 0.08
## ratings_post               4 26  3.04 0.52 0.10
## ratings_change_scores      5 26  0.12 0.40 0.08
## D1                         6 26 -0.08 0.38 0.07
##
## attr(,"call")
## by.data.frame(data = x, INDICES = group, FUN = describe, type = type,
##     fast = TRUE, ranges = FALSE, trim = 0)
```

# Distribution plots

Ratings change scores

```
ggplot(data = data_df, aes(x = ratings_change_scores, colour = condition, fill = condition)) +
  geom_density(alpha = 0.3) +
  theme_minimal()
```




# Manipulation checks

## Differences in ratings of valenced stimuli

### Frequentist

T test

```
t.test(formula = valenced_stimuli_ratings ~ condition,
       data = data_df, alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  valenced_stimuli_ratings by condition
## t = 14.523, df = 50, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  2.034584      Inf
## sample estimates:
## mean in group flowers mean in group insects
##              4.261538              1.961538
```

```
cohen.d(formula = valenced_stimuli_ratings ~ condition,
        data = data_df)
```

```
##
## Cohen's d
##
## d estimate: 4.027894 (large)
## 95 percent confidence interval:
##      inf      sup
## 3.026171 5.029617
```

### Bayes factors

```
ttestBF(formula = valenced_stimuli_ratings ~ condition,
        nullInterval = c(-Inf,0),
        rscale = "medium",  # i.e., r = .707 # directional hypothesis
        data = data_df)
```

```
## t is large; approximation invoked.
## t is large; approximation invoked.
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 -Inf<d<0    : 0.002744922  ±NA%
## [2] Alt., r=0.707 !(-Inf<d<0) : 3.829152e+16 ±NA%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

## Differences in (training) IAT effects betwen conditions

### Frequentist

T test

```
t.test(formula = D1 ~ condition,
       data = data_df,
       alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  D1 by condition
## t = -2.2382, df = 49.817, p-value = 0.01486
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##         -Inf -0.06134149
## sample estimates:
## mean in group flowers mean in group insects
##           -0.32615385           -0.08192308
```

```
cohen.d(formula = D1 ~ condition,
        data = data_df)
```

```
##
## Cohen's d
```

```
##
## d estimate: -0.6207546 (medium)
## 95 percent confidence interval:
##         inf         sup
## -1.20291635 -0.03859292
```

#### Bayes factors

```
ttestBF(formula = D1 ~ condition,
        data = data_df,
        rscale = "medium",  # i.e., r = .707
        nullInterval = c(-Inf,0)) # directional hypothesis
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 -Inf<d<0    : 4.08138    ±0%
## [2] Alt., r=0.707 !(-Inf<d<0) : 0.09713165 ±0.01%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

## Hypothesis tests

### Differences in ratings changes between conditions

#### Frequentist

T test

```
# t test
t.test(ratings_change_scores ~ condition,
       data = data_df, alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  ratings_change_scores by condition
## t = -3.1351, df = 46.262, p-value = 0.001491
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.1894193
## sample estimates:
## mean in group flowers mean in group insects
##            -0.2923077             0.1153846
```

```
# effect size
cohen.d(ratings_change_scores ~ condition,
        data = data_df)
```

```
##
## Cohen's d
##
```

```
## d estimate: -0.869507 (large)
## 95 percent confidence interval:
##        inf        sup
## -1.4648774 -0.2741366
```

Alternative strategy: ANCOVA with pre as covariate

```
model1 <- lm(ratings_post ~ ratings_pre + condition,
             data = data_df)

ratings_ANCOVA <-
  etaSquared(model1,
             type = 3,
             anova = TRUE) %>% # output full anova results, not just eta2
  as.data.frame()

ratings_ANCOVA
```

```
##                eta.sq eta.sq.part        SS df        MS        F
## ratings_pre 0.3200819   0.3790006  6.286161  1 6.2861615 29.90506
## condition   0.1197851   0.1859310  2.352488  1 2.3524878 11.19145
## Residuals   0.5244601          NA 10.299992 49 0.2102039       NA
##                        p
## ratings_pre 1.524150e-06
## condition   1.583157e-03
## Residuals             NA
```

```
# 90% CI on eta2
## (nb 90% not 95%, see Wuensch, 2009; Steiger. 2004)
# from http://daniellakens.blogspot.be/2014/06/calculating-confidence-intervals-for.html
# 1. extract individual stats
ancova_F        <-  ratings_ANCOVA$F[2]        # where 2 specifies the main effect row
ancova_df_1     <-  ratings_ANCOVA$df[2]       # where 2 specifies the main effect row
ancova_df_2     <-  ratings_ANCOVA$df[3]       # where 3 specifies the residuals row
ancova_p        <-  ratings_ANCOVA$p[2]        # where 2 specifies the main effect row
ancova_eta2     <-  ratings_ANCOVA$eta.sq[2]   # where 2 specifies the main effect row

n_df <- data_df %>% dplyr::summarize(n_variable = n())
n_integer <- n_df$n_variable

# 2. Use to calculate 90% CIs
ci.pvaf(F.value = ancova_F,
        df.1 = ancova_df_1,
        df.2 = ancova_df_2,
        N = n_integer,
        conf.level=.90)
```

```
## $Lower.Limit.Proportion.of.Variance.Accounted.for
## [1] 0.04656081
##
## $Probability.Less.Lower.Limit
## [1] 0.05
##
## $Upper.Limit.Proportion.of.Variance.Accounted.for
## [1] 0.330375
##
```

```
## $Probability.Greater.Upper.Limit
## [1] 0.05
##
## $Actual.Coverage
## [1] 0.9
```

### Bayes factors

```
ttestBF(formula = ratings_change_scores ~ condition,
        rscale = "medium",  # i.e., r = .707
        nullInterval = c(-Inf,0), # directional hypothesis
        data = data_df)
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 -Inf<d<0    : 25.98621   ±0%
## [2] Alt., r=0.707 !(-Inf<d<0) : 0.07743824 ±0.13%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

## Post hoc exploratory tests

Added at the request of our peer reviewers.

### Is the effect influenced by IAT block order?

Explore interaction effect between condition and block order.

### Frequentist

```
# check factors are indeed set to factors
sapply(data_df, class)
```

```
##              participant                  gender                     age
##                "integer"                "factor"               "integer"
##             IAT_accuracy             block_order                      D1
##                "numeric"                "factor"               "numeric"
##                condition             IAT_mean_RT             ratings_pre
##                 "factor"               "integer"               "numeric"
##             ratings_post valenced_stimuli_ratings   ratings_change_scores
##                "numeric"               "numeric"               "numeric"
```

```
model2 <-
  lm(ratings_change_scores ~ condition * block_order,
     contrasts = list(condition = "contr.sum", block_order = "contr.sum"),  # effect coding for factor v
     data = data_df)

etaSquared(model2,
```

```
          type = 3,
          anova = TRUE)
```

```
##                             eta.sq  eta.sq.part           SS df
## condition             1.642786e-01 0.1642978300 2.160769e+00  1
## block_order           5.848295e-05 0.0000699839 7.692308e-04  1
## condition:block_order 5.848295e-05 0.0000699839 7.692308e-04  1
## Residuals             8.356044e-01           NA 1.099077e+01 48
##                                 MS           F           p
## condition             2.1607692308 9.436730123 0.003498669
## block_order           0.0007692308 0.003359462 0.954020420
## condition:block_order 0.0007692308 0.003359462 0.954020420
## Residuals             0.2289743590          NA          NA
```

### Bayes factors

```
anovaBF(formula = ratings_change_scores ~ condition * block_order,
        data = data_df,
        rscaleFixed = "medium",
        multicore = TRUE)
```

```
## Note: Progress bars and callbacks are suppressed when running multicore.
```

```
## Bayes factor analysis
## --------------
## [1] block_order                                     : 0.278549 ±0.02%
## [2] condition                                       : 13.03182 ±0%
## [3] block_order + condition                         : 3.652007 ±0.81%
## [4] block_order + condition + block_order:condition : 1.251991 ±2.95%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```