R Notebook

Josh Archer, Hannah Choi, Josh Lin, Pauline Yue

All About Attraction: Does Working at Trader Joe's Increase Your Attractiveness?

Abstract

Background

Over the years, Trader Joe's has become more than just a place to get groceries, but rather an experience in itself. From tropical island themed interior and uniforms for their employees, to unique products only found at Trader Joe's, the brand has a competitive edge in the grocery industry. The culture around the store has allowed it to develop a cult following of customers who rave about their products. More recently, there has been increasing attention towards the store's employees and the general opinion that Trader Joe's workers are more attractive than the average individual. Based on this observed social phenomenon, we conducted an experiment to test this hypothesis.

Research Question & Hypothesis

Does working at Trader Joe's make you more attractive?

Experiment/Research Design

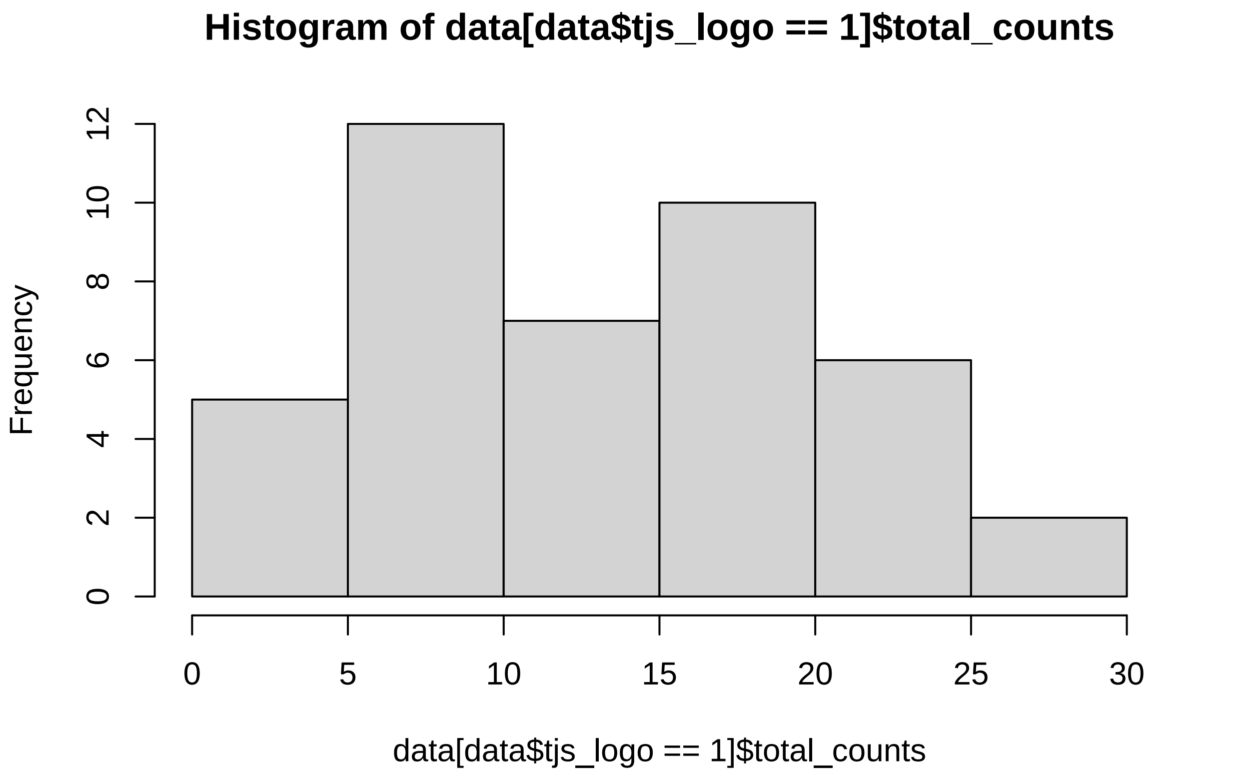
Study Design

For our study, we implemented a swipe-style survey similar to modern-day dating apps like Tinder, in which participants swipe through various photos and choose the individual they are more attracted to. We obtained photos from our personal circle of friends, and we attempted to maintain a wide array of individuals in our photo sample. In total, we received 30 photos from 15 female-identifying folks, and 36 photos from 19 male-identifying folks. We also designed a pilot study similar to our experimental survey that featured the same swipe design, but on a smaller scale. We distributed the pilot study first to determine how long it would take participants to finish the survey and what feedback they might give in response to our free response question to help us improve our final iteration. We then posted our final survey through Amazon Mturk.

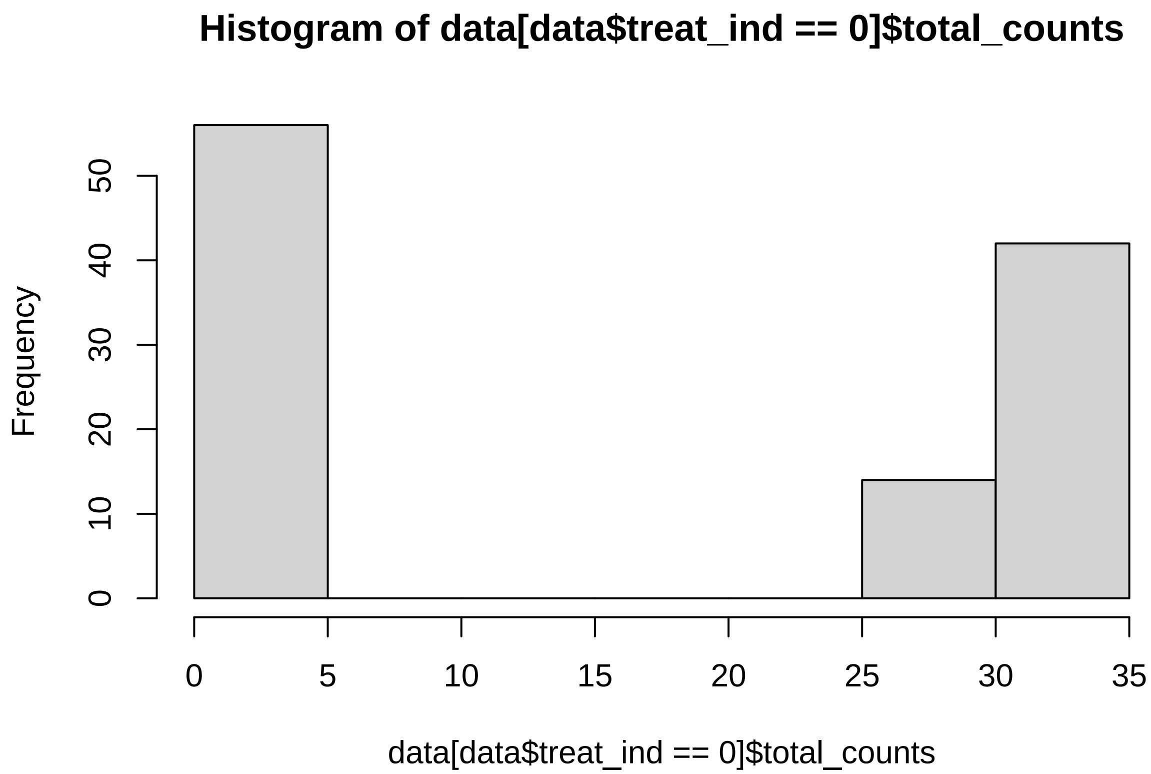
```r
data <- fread("../data/processed/analysis_table.csv")

data[, treat_ind := ifelse(treat=='t',1,0)]

data[, treat_with_logo := ifelse((logo != 'none' & treat == 't'),1 ,0)]

data[, tjs_logo := ifelse(test = grepl(pattern = "tjs", data[, logo]),
                                              yes = 1, no = 0)]

data[, nike_logo := ifelse(test = grepl(pattern = "nike", data[, logo]),
                                              yes = 1, no = 0)]

data[, no_logo := ifelse(test = grepl(pattern = "none", data[, logo]),
```

```
                                                    yes = 1, no = 0)]
data[, survey_block := gsub("^.*_", "", data$photo_block)]

#names(data)

#View(data)
```

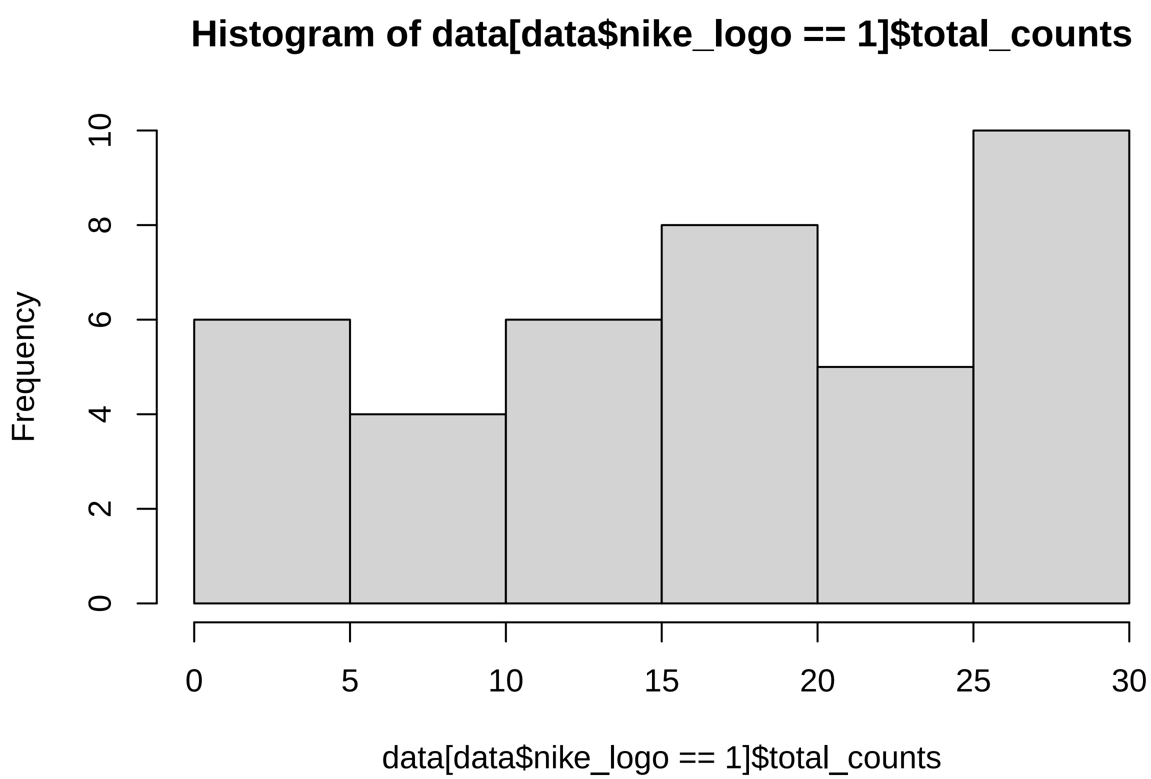
```
hist(data[data$tjs_logo==1]$total_counts)
```



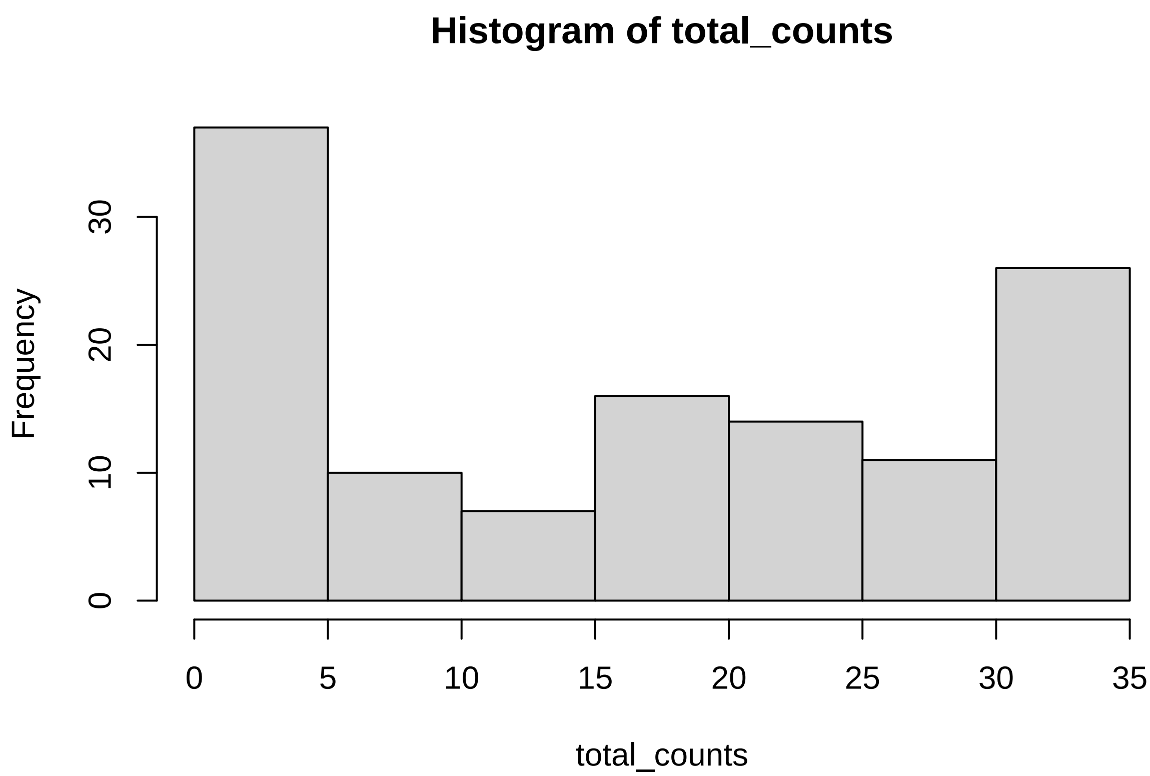

```
hist(data[data$treat_ind==0]$total_counts)
```




```r
hist(data[data$nike_logo==1]$total_counts)
```

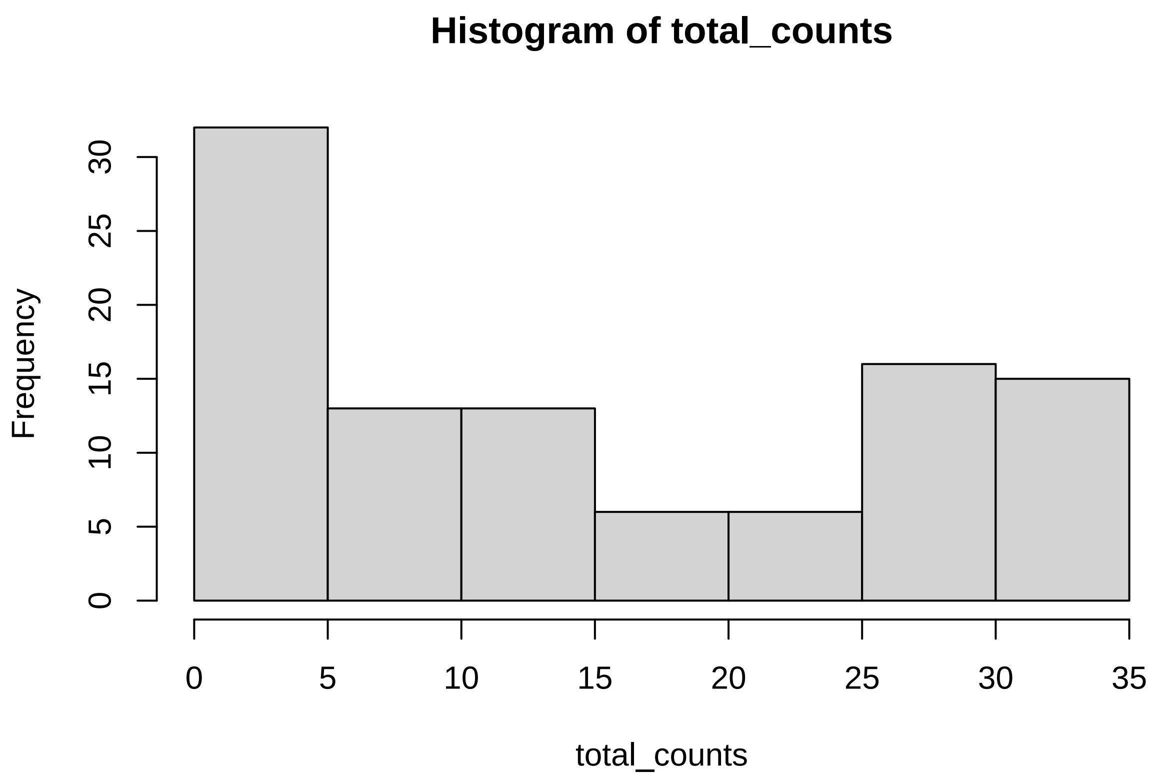



```r
data[smile == 'smile', hist(total_counts)]
```




```
## $breaks
## [1]  0  5 10 15 20 25 30 35
##
## $counts
## [1] 37 10  7 16 14 11 26
##
## $density
## [1] 0.06115702 0.01652893 0.01157025 0.02644628 0.02314050 0.01818182 0.04297521
##
## $mids
## [1]  2.5  7.5 12.5 17.5 22.5 27.5 32.5
##
## $xname
## [1] "total_counts"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
data[smile == 'nonsmile', hist(total_counts)]
```




```
## $breaks
## [1]  0  5 10 15 20 25 30 35
##
## $counts
## [1] 32 13 13  6  6 16 15
##
## $density
## [1] 0.06336634 0.02574257 0.02574257 0.01188119 0.01188119 0.03168317 0.02970297
##
## $mids
## [1]  2.5  7.5 12.5 17.5 22.5 27.5 32.5
##
## $xname
## [1] "total_counts"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
robust_se <- function(mod, type = 'HC3') { sqrt(diag(vcovHC(mod, type)))}
```

```
model_1 <- data[, lm(total_counts ~ tjs_logo)]
coeftest(model_1, vcovHC(model_1))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  15.8022     0.9887 15.9828   <2e-16 ***
## tjs_logo     -1.9450     1.5123 -1.2862   0.1997
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_2 <- data[, lm(total_counts ~ tjs_logo + survey_block)]
coeftest(model_2, vcovHC(model_2))
```

```
##
## t test of coefficients:
##
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   15.78211    1.68154  9.3855   <2e-16 ***
## tjs_logo      -1.97474    1.54665 -1.2768   0.2030
## survey_blockb  0.64105    2.35629  0.2721   0.7858
## survey_blockc -0.42947    2.39987 -0.1790   0.8581
## survey_blockd -0.10895    2.32901 -0.0468   0.9627
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model_1, model_2, test= 'F')
```

```
## Analysis of Variance Table
##
## Model 1: total_counts ~ tjs_logo
## Model 2: total_counts ~ tjs_logo + survey_block
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    222 34226
## 2    219 34192  3    33.829 0.0722 0.9748
```

Although the F-test p-value is almost 1, looking at this second baseline model, we can see that our randomization was successful, since we did not see within-survey group effects. In other words, our treatment fulfilled the exclusion principle. (*add more here later*)

```
model_3 <- data[nike_logo ==  0, lm(total_counts ~ tjs_logo)]
coeftest(model_3, vcovHC(model_3))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  15.5175     1.1853 13.0919   <2e-16 ***
## tjs_logo     -1.6603     1.6475 -1.0078   0.3149
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_4 <- data[, lm(total_counts ~ tjs_logo + smile)]
coeftest(model_4, vcovHC(model_4))
```

```
##
## t test of coefficients:
##
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)     28.5000     3.5355  8.0610 4.806e-14 ***
## tjs_logo        -1.7760     1.5244 -1.1650 0.2452661
## smilenonsmile -13.4255     3.7865 -3.5456 0.0004785 ***
## smilesmile    -12.3589     3.7523 -3.2937 0.0011521 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(model_1, model_4, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: total_counts ~ tjs_logo
## Model 2: total_counts ~ tjs_logo + smile
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    222 34226
## 2    220 33837  2    388.66 1.2635 0.2847
```

(Insert stargazer here for initial model creation, 1-4)

```r
# Remove nike logo photos and look for just tjs photos
interim <- data[logo != "nike"]
tjs_only <- data[logo == "tjs"]

# Create a join_id to combine tables later
photo_name_split <- read.table(text = interim[ , photo_block], sep = "_", as.is = TRUE, fill = TRUE)
interim[, join_id := paste(photo_name_split$V1, photo_name_split$V2, photo_name_split$V3, photo_name_spl
interim[, gender := photo_name_split$V2]

## Subset needed columns and match control photos to tjs photos (not including photos that were blank i
tjs_only <- interim[logo=="tjs"]
tjs_only <- tjs_only[ , c("photo_block", "join_id", "treat_ind", "survey_block", "total_counts", "smile
rows_to_keep <- tjs_only[, join_id]
none_c_only <- interim[logo=="none" & treat_ind<1]
none_c_only <- none_c_only[ , c("photo_block", "join_id", "treat_ind", "survey_block", "total_counts",
none_c_matched <- subset(none_c_only, join_id %in% rows_to_keep)

# none_c_rows <- none_c_matched[, join_id]
# tjs_issue <- subset(tjs_only, !(join_id %in% none_c_rows))

# View(none_c_matched)
tjs_vs_control <- rbind(tjs_only, none_c_matched)
#View(tjs_vs_control)
```

```r
#42 blanks and 42 with tj's
model_5 <- tjs_vs_control[, lm(total_counts ~ treat_ind)]
coeftest(model_5, vcovHC(model_5))
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  17.7143     2.4269  7.2992 1.676e-10 ***
## treat_ind    -3.8571     2.6831 -1.4376    0.1544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_6 <- tjs_vs_control[, lm(total_counts ~ treat_ind + smile + treat_ind*smile)]
coeftest(model_6, vcovHC(model_6))
```

```
##
## t test of coefficients:
```

```
##
##                        Estimate Std. Error t value  Pr(>|t|)
## (Intercept)             16.8500     3.5900  4.6936 1.094e-05 ***
## treat_ind               -2.8500     4.0518 -0.7034    0.4839
## smilesmile               1.6500     4.9723  0.3318    0.7409
## treat_ind:smilesmile    -1.9227     5.5094 -0.3490    0.7280
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model_5, model_6, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: total_counts ~ treat_ind
## Model 2: total_counts ~ treat_ind + smile + treat_ind * smile
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     82 12102
## 2     80 12072  2    29.301 0.0971 0.9076
```

```
model_7 <- tjs_vs_control[, lm(total_counts ~ treat_ind + gender + treat_ind*gender)]
coeftest(model_7, vcovHC(model_7))
```

```
##
## t test of coefficients:
##
##                     Estimate Std. Error t value  Pr(>|t|)
## (Intercept)          17.1000     3.6433  4.6935 1.094e-05 ***
## treat_ind            -4.9500     4.0504 -1.2221    0.2253
## genderm               1.1727     4.9825  0.2354    0.8145
## treat_ind:genderm     2.0864     5.4893  0.3801    0.7049
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model_5, model_7, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: total_counts ~ treat_ind
## Model 2: total_counts ~ treat_ind + gender + treat_ind * gender
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     82 12102
## 2     80 11976  2    125.68 0.4198 0.6586
```

```
model_gen_smile <- tjs_vs_control[, lm(total_counts ~ treat_ind + smile + treat_ind*smile
                                       + gender + treat_ind*gender + smile*gender)]
coeftest(model_gen_smile, vcovHC(model_gen_smile))
```

```
##
## t test of coefficients:
##
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             18.1404     5.0406  3.5988 0.000563 ***
## treat_ind               -3.9474     5.2947 -0.7455 0.458219
## smilesmile              -1.8915     6.0410 -0.3131 0.755039
## genderm                 -2.3461     5.9870 -0.3919 0.696243
## treat_ind:smilesmile    -1.8230     5.6490 -0.3227 0.747793
## treat_ind:genderm        1.9952     5.6430  0.3536 0.724622
```

```
## smilesmile:genderm     6.8485     5.6725  1.2073 0.231006
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model_5, model_gen_smile, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: total_counts ~ treat_ind
## Model 2: total_counts ~ treat_ind + smile + treat_ind * smile + gender +
##     treat_ind * gender + smile * gender
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     82 12102
## 2     77 11701  5    400.85 0.5276 0.7547
```

```
stargazer(model_5, model_6, model_7, model_gen_smile, type = 'text',
          se = list(robust_se(model_5), robust_se(model_6), robust_se(model_7),
                    robust_se(model_gen_smile)))
```

```
##
## ==================================================================================================
##                                                Dependent variable:
##                       ----------------------------------------------------------------------------
##                                                   total_counts
##                              (1)                 (2)                 (3)                 (4)
## --------------------------------------------------------------------------------------------------
## treat_ind                  -3.857              -2.850              -4.950              -3.947
##                            (2.683)             (4.052)             (4.050)             (5.295)
##
## smilesmile                                      1.650                                  -1.892
##                                                (4.972)                                 (6.041)
##
## treat_ind:smilesmile                           -1.923                                  -1.823
##                                                (5.509)                                 (5.649)
##
## genderm                                                             1.173              -2.346
##                                                                    (4.983)             (5.987)
##
## treat_ind:genderm                                                   2.086               1.995
##                                                                    (5.489)             (5.643)
##
## smilesmile:genderm                                                                      6.848
##                                                                                        (5.672)
##
## Constant                  17.714***           16.850***           17.100***           18.140***
##                            (2.427)             (3.590)             (3.643)             (5.041)
##
## --------------------------------------------------------------------------------------------------
## Observations                 84                  84                  84                  84
## R2                          0.025               0.028               0.035               0.057
## Adjusted R2                 0.013              -0.009              -0.001              -0.016
## Residual Std. Error   12.148 (df = 82)    12.284 (df = 80)    12.235 (df = 80)    12.327 (df = 77)
## F Statistic          2.117 (df = 1; 82) 0.755 (df = 3; 80) 0.976 (df = 3; 80) 0.782 (df = 6; 77)
## ==================================================================================================
## Note:                                                                 *p<0.1; **p<0.05; ***p<0.01
```

```r
interim <- copy(data)
photo_name_split <- read.table(text = interim[ , photo_block], sep = "_", as.is = TRUE, fill = TRUE)
interim[, join_id := paste(photo_name_split$V1, photo_name_split$V2, photo_name_split$V3, photo_name_spl
interim[, gender := photo_name_split$V2]
# View(interim)

nike_only <- interim[logo=="nike"]
nike_only <- nike_only[ , c("photo_block", "join_id", "treat_ind", "survey_block", "total_counts", "smil
rows_to_keep_nike <- nike_only[, join_id]
none_c_matched <- subset(none_c_only, join_id %in% rows_to_keep_nike)
# none_c_rows <- none_c_matched[, join_id]
# tjs_issue <- subset(tjs_only, !(join_id %in% none_c_rows))
# View(none_c_matched)
nike_vs_control <- rbind(nike_only, none_c_matched) # only has 39 blanks and 39 nike logo photos
```

```r
model_8 <- nike_vs_control[, lm(total_counts ~ treat_ind)]
coeftest(model_8, vcovHC(model_8))
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  11.8718     2.4687  4.8089 7.503e-06 ***
## treat_ind     4.9744     2.9355  1.6945   0.09426 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To ensure that the effect we observed for our above Trader Joe's baseline model (Model 5) is not due to simply adding a logo to a photo, we included photos that had a Nike logo as a robustness check. While the number of Nike logo pairs is smaller (n=39), the overall ratio is almost the same, and because of the similar proportion of photos pairs, we can infer that the estimated effect we saw in the Trader Joe's baseline model is unique to the Trader Joe's logo as they two slope coefficients are vastly different.

```r
#interaction term not included because of perfect collinearity (potentially a product of randomization)
model_9 <- nike_vs_control[, lm(total_counts ~ treat_ind + smile)]
coeftest(model_9, vcovHC(model_9))
```

```
##
## t test of coefficients:
##
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)     26.0128     7.3863  3.5217 0.0007383 ***
## treat_ind        4.9744     2.9404  1.6917 0.0949061 .
## smilenonsmile -15.4286     7.6543 -2.0157 0.0474656 *
## smilesmile    -13.9792     7.4770 -1.8696 0.0654940 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(model_8, model_9, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: total_counts ~ treat_ind
## Model 2: total_counts ~ treat_ind + smile
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     76 12443
```

```
## 2     74 11996  2    447.61 1.3806 0.2578
```

```r
model_10 <- nike_vs_control[, lm(total_counts ~ treat_ind + gender + treat_ind*gender)]
coeftest(model_10, vcovHC(model_10))
```

```
##
## t test of coefficients:
##
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         11.36842    3.60573  3.1529 0.002336 **
## treat_ind            6.10526    4.46701  1.3667 0.175844
## genderm              0.98158    5.06796  0.1937 0.846955
## treat_ind:genderm -2.20526    6.03723 -0.3653 0.715946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(model_8, model_10, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: total_counts ~ treat_ind
## Model 2: total_counts ~ treat_ind + gender + treat_ind * gender
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     76 12443
## 2     74 12420  2    23.978 0.0714 0.9311
```

```r
model_nike_gen_smile <- nike_vs_control[, lm(total_counts ~ treat_ind + smile
                                       + gender + treat_ind*gender)]
coeftest(model_nike_gen_smile, vcovHC(model_nike_gen_smile))
```

```
##
## t test of coefficients:
##
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          25.4474     8.5807  2.9657 0.004095 **
## treat_ind             6.1053     4.4375  1.3758 0.173138
## smilenonsmile       -15.7314     8.8289 -1.7818 0.078997 .
## smilesmile          -14.3073     8.6840 -1.6475 0.103805
## genderm               1.7084     5.1079  0.3345 0.739010
## treat_ind:genderm    -2.2053     6.0550 -0.3642 0.716774
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(model_8, model_nike_gen_smile, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: total_counts ~ treat_ind
## Model 2: total_counts ~ treat_ind + smile + gender + treat_ind * gender
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     76 12443
## 2     72 11965  4    478.25 0.7195 0.5814
```

```r
stargazer(model_8, model_9, model_10, model_nike_gen_smile, type = 'text',
          se = list(robust_se(model_8), robust_se(model_9), robust_se(model_10),
                    robust_se(model_nike_gen_smile)))
```

```
##
```

```
## ================================================================================
##                                     Dependent variable:
##                    ------------------------------------------------------------
##                                         total_counts
##                         (1)            (2)            (3)            (4)
## --------------------------------------------------------------------------------
## treat_ind             4.974*         4.974*         6.105          6.105
##                      (2.936)        (2.940)        (4.467)        (4.438)
##
## smilenonsmile                      -15.429**                     -15.731*
##                                     (7.654)                       (8.829)
##
## smilesmile                         -13.979*                      -14.307*
##                                     (7.477)                       (8.684)
##
## genderm                                            0.982          1.708
##                                                   (5.068)        (5.108)
##
## treat_ind:genderm                                 -2.205         -2.205
##                                                   (6.037)        (6.055)
##
## Constant             11.872***      26.013***      11.368***      25.447***
##                      (2.469)        (7.386)        (3.606)        (8.581)
##
## --------------------------------------------------------------------------------
## Observations            78             78             78             78
## R2                     0.037          0.072          0.039          0.074
## Adjusted R2            0.025          0.034          0.0002         0.010
## Residual Std. Error 12.796 (df = 76)  12.732 (df = 74)  12.955 (df = 74)  12.891 (df = 72)
## F Statistic        2.947* (df = 1; 76) 1.913 (df = 3; 74) 1.006 (df = 3; 74) 1.156 (df = 5; 72)
## ================================================================================
## Note:                                             *p<0.1; **p<0.05; ***p<0.01
```

Power Calculation

```r
#Power calculation using model_6
summary(model_5)
```

```
##
## Call:
## lm(formula = total_counts ~ treat_ind)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.714  -9.357   1.143  12.286  16.143
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   17.714      1.875   9.450 9.17e-15 ***
## treat_ind     -3.857      2.651  -1.455    0.149
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.15 on 82 degrees of freedom
```

```
## Multiple R-squared:  0.02517,    Adjusted R-squared:  0.01328
## F-statistic: 2.117 on 1 and 82 DF,  p-value: 0.1495
```

```
effect_size <- 0.02517/(1-0.02517) #use multiple R-squared value from model 6 summary
pwr.f2.test(u = 1, v = 82, f2 = effect_size) #u and v come from DF from summary
```

```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 82
##              f2 = 0.02581989
##       sig.level = 0.05
##           power = 0.3072778
```

To determine how much statistical power our final model (Model 5) had, based on our model parameters, we found that our model had about 30.73% statistical power. This means that our model only had 30.73% chance of reporting a true positive result. Although the reported statistical power is quite low, we feel that this level of power is understandable given our small data size and the fact we were dealing with data related to how humans think.

First, when we only focused on photos that were blank in control and contained a Trader Joe's logo in treatment, we only had 42 total unique photo subjects. Therefore, we were comparing across 42 pairs, which is a fairly small dataset. Second, when dealing with sociology- and psychology-type questions like attraction, given the limitations of our experimental design, we cannot fully capture how humans think using the variables we had operationalized. Therefore, we argue that it is reasonable that our statistical power is quite small.

```
#where should this go/should we keep it
#tjs_vs_nike <- data[, t.test(data[tjs_logo == 1, total_counts], data[nike_logo == 1, total_counts])]
#tjs_vs_nike
#tjs_vs_control <- data[, t.test(data[tjs_logo == 1, total_counts], data[tjs_logo == 0, total_counts], p
#tjs_vs_control
#nike_vs_control <- data[, t.test(data[nike_logo == 1, total_counts], data[nike_logo == 0, total_counts
#nike_vs_control
#smile_vs_nonsmile <- data[, t.test(data[smile == 'smile', total_counts], data[smile == 'nonsmile', tot
#smile_vs_nonsmile
```