Feedback MTurk Study

Introduction

Load Data

```
# d <- fread('Lungs_November+14,+2020_17.33.csv')
d <- fread('../check-valid-responses/data/qualtrics_results_final.csv')
#head(d)
```

```
d_respondents_only <-
  d[(Status == "IP Address") & (Finished == 'True'),]

# Remove these survey responses because they were from people who did the survey again. Double check th
d_respondents_only <-
  d_respondents_only[!ResponseId %in% c(
    'R_1eRkKqfVAmkVzj2',
    'R_3FRO3xu5zyOsRSU',
    'R_3HBQsMSMCgXPpKf',
    'R_dbzictBknL9jG3T'
  ), ]

# These WorkerId put in all 1 response (all Normal or all Pneumonia)
d_respondents_only <-
  d_respondents_only[!Q80 %in% c(
    "A119EX2LODNN1B",
    "A12NQJV6TA5OWB",
    "A18WFPSLFV4FKY",
    "A1BUYK6LXYWMLL",
    "A1FHRZXSE7XNJ4",
    "A1GMYDH5MKN1O5",
    "A2GSZ3D2XXC533",
    "A2IGIOD74EPOEF",
    "A2JO16DRTOBXWO",
    "A2NGFU82LMJ8OX",
    "A32K1MOA36EAK5",
    "A371SNJNNUY9Z6",
    "A3BPENSX5EVJ2H",
    "A3EPIT2P3ISA3K",
    "A3NYIJYBHAJ74V",
    "AUFLTHQAXWLH1",
    "AVINXZZV3FNG7" ,
    "A1CD7O6OQAQQRT",
    "A1CF1W8CPODHBO",
    "A1PGY59BR6C5BX",
    "A1YSYI926BBOHW",
    "A1Z3GFH6MNSU46",
    "A211KGJ94WNFLN",
```

```
    "A26RPQDD0RQEHL",
    "A2BUHMLNE3LUU0",
    "A2J5BRQ88W745H",
    "A2XIHO2W7EEP32",
    "A3EZ0H07TSDAPW",
    "A3FLBC6LC5GJ3W",
    "A3QLKLIQW1B1FR",
    "A8F6JFG0WSELT",
    "A9K6IVBA0J1CX",
    "ADLZLGHK0AEE6",
    "AE7NJG0K0VZYJ" ,
    "AG5RF4UGQJ7A7" ,
    "AQ9Y6WD8O72ZC" ,
    "tuturtu"
  ), ]

# These people just gave alternating responses (Normal, Pneumonia, Normal,...,Pneumonia)

d_respondents_only <- d_respondents_only[!Q80 %in% c(
'A1W05TSPORJPXR'
,'A3SUWCLD1GEGM7'
,'A3A09JB9X1RBXW'
,'A7VQQEIBSM9IU'
,'A8DER1QY96C5X'
,'A1M8MNKK8H5ZGW'
,'A34D5D6PU193AR'
),]

#head(d_respondents_only)
```

```
#rename task phase questions
setnames(d_respondents_only,
         old = c('Q2', 'Q42'),
         new = c('Self_Reflect_Q1', 'Self_Reflect_Q2'))

setnames(d_respondents_only,
         old = c('Q69', 'Q89'),
         new = c('Control_Q1', 'Control_Q2'))

setnames(d_respondents_only,
         old = c('Q80', 'Q82', 'Q83', 'Q84', 'SC0', 'FL_6_DO'),
         new = c('Amazon_Turk_ID', 'Gender', 'Age_Range', 'Education_Level', 'Total_Score', 'Assignment

setnames(d_respondents_only,
         old = c('Q1', 'Q5', 'Q6', 'Q7', 'Q16', 'Q17', 'Q18', 'Q19', 'Q20', 'Q21',
                 'Q8', 'Q9', 'Q10', 'Q11', 'Q22', 'Q23', 'Q24', 'Q25', 'Q26', 'Q27',
                 'Q12', 'Q13', 'Q14', 'Q15', 'Q28', 'Q29', 'Q30', 'Q31', 'Q32', 'Q33'),
         new = c('Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'Q6', 'Q7', 'Q8', 'Q9', 'Q10',
                 'Q11', 'Q12', 'Q13', 'Q14', 'Q15', 'Q16', 'Q17', 'Q18', 'Q19', 'Q20',
                 'Q21', 'Q22', 'Q23', 'Q24', 'Q25', 'Q26', 'Q27', 'Q28', 'Q29', 'Q30'))

d_respondents_only[ , c("Q1_Score", "Q2_Score", "Q3_Score", "Q4_Score", "Q5_Score",
                        "Q6_Score", "Q7_Score", "Q8_Score", "Q9_Score", "Q10_Score",
                        "Q11_Score", "Q12_Score", "Q13_Score", "Q14_Score", "Q15_Score",
```

```
                        "Q16_Score", "Q17_Score", "Q18_Score", "Q19_Score", "Q20_Score",
                        "Q21_Score", "Q22_Score", "Q23_Score", "Q24_Score", "Q25_Score", "Q26_Score",
                        "Q27_Score", "Q28_Score", "Q29_Score", "Q30_Score") :=
                      list(ifelse(Q1 == "Normal", 1, 0),
                           ifelse(Q2 == "Normal", 1, 0),
                           ifelse(Q3 == "Pneumonia", 1, 0),
                           ifelse(Q4 == "Pneumonia", 1, 0),
                           ifelse(Q5 == "Normal", 1, 0),
                           ifelse(Q6 == "Pneumonia", 1, 0),
                           ifelse(Q7 == "Pneumonia", 1, 0),
                           ifelse(Q8 == "Normal", 1, 0),
                           ifelse(Q9 == "Pneumonia", 1, 0),
                           ifelse(Q10 == "Normal", 1, 0),
                           ifelse(Q11 == "Pneumonia", 1, 0),
                           ifelse(Q12 == "Normal", 1, 0),
                           ifelse(Q13 == "Pneumonia", 1, 0),
                           ifelse(Q14 == "Pneumonia", 1, 0),
                           ifelse(Q15 == "Normal", 1, 0),
                           ifelse(Q16 == "Normal", 1, 0),
                           ifelse(Q17 == "Pneumonia", 1, 0),
                           ifelse(Q18 == "Normal", 1, 0),
                           ifelse(Q19 == "Pneumonia", 1, 0),
                           ifelse(Q20 == "Normal", 1, 0),
                           ifelse(Q21 == "Normal", 1, 0),
                           ifelse(Q22 == "Normal", 1, 0),
                           ifelse(Q23 == "Pneumonia", 1, 0),
                           ifelse(Q24 == "Normal", 1, 0),
                           ifelse(Q25 == "Pneumonia", 1, 0),
                           ifelse(Q26 == "Pneumonia", 1, 0),
                           ifelse(Q27 == "Pneumonia", 1, 0),
                           ifelse(Q28 == "Pneumonia", 1, 0),
                           ifelse(Q29 == "Normal", 1, 0),
                           ifelse(Q30 == "Normal", 1, 0))]

d_respondents_only[ , Assignment_Group := ifelse(Assignment == "FL_17", "Control",
                                          ifelse(Assignment == "FL_14", "Self-Reflect",
                                          ifelse(Assignment == "FL_15", "Medical Feedback",
                                          ifelse(Assignment == "FL_16", "Positive Images", "Negative Ima

d_respondents_only[ , c("TaskPhase1_Score", "TaskPhase2_Score", "TaskPhase3_Score") :=
                      list(sum(Q1_Score, Q2_Score, Q3_Score, Q4_Score, Q5_Score, Q6_Score, Q7_Score, Q8
                           sum(Q11_Score, Q12_Score, Q13_Score, Q14_Score, Q15_Score, Q16_Score, Q17_Sc
                           sum(Q21_Score, Q22_Score, Q23_Score, Q24_Score, Q25_Score, Q26_Score, Q27_Sc
                    by = Amazon_Turk_ID]

#head(d_respondents_only)
```

```
# ?register_google
# register_google(key = "AIzaSyCTk2a5vIEqcvgz9KmQmItoNF7J8_hiMMk")
#
# #uses Google API to obtain location data based on longitude and latitude....dont use unless necessary
# d_respondents_only[ , c("housenumber", "street", "city", "county", "state", "zip", "country") := revg
# #
# head(d_respondents_only)
```

```
# #
# #
# fwrite(d_respondents_only, file='datatable_clean_survey_responses_v2.dta')

d_respondents <- fread('datatable_clean_survey_responses_v2.dta')


#head(d_respondents)
```

```
nrow(d_respondents)
```

```
## [1] 350
```

```
#skip

# ?register_google
# register_google(key = "AIzaSyCTk2a5vIEqcvgz9KmQmItoNF7J8_hiMMk")
# ggmap_show_api_key()
#
#
# revgeocode(c(df$lon[1], df$lat[1]))
#
# d_respondents_only[ Q80 == "A1AC47WJLNW4G7", revgeocode(c(as.numeric(LocationLongitude)[1], as.numeri
# ?revgeocode
```

```
#remove duplicate Amazon Turk IDs
nrow(d_respondents)  #350 rows
```

```
## [1] 350
```

```
d_respondents <- d_respondents[ !duplicated(d_respondents$Amazon_Turk_ID) , ]  #350 rows
```

EDA

```
#some EDA

#d_respondents[ , table(state, country)]


table(d_respondents$state, d_respondents$country) %>%
        as.data.frame() %>%
        arrange(desc(Freq)) %>%
        filter(Freq>0)
```

```
##                         Var1           Var2 Freq
## 1                 Tamil Nadu          India  107
## 2                 California  United States   72
## 3                   New York  United States   22
## 4                     Kansas  United States   21
## 5                      Texas  United States   15
## 6                    Florida  United States    9
## 7              Massachusetts  United States    7
## 8                   Missouri  United States    6
## 9                Connecticut  United States    5
## 10                   Georgia  United States    5
```

```
## 11                   Indiana  United States    5
## 12                  Michigan  United States    5
## 13                New Jersey  United States    5
## 14                  Illinois  United States    4
## 15                  Virginia  United States    4
## 16                    Kerala          India    3
## 17               Maharashtra          India    3
## 18                  Colorado  United States    3
## 19                  Kentucky  United States    3
## 20                  Maryland  United States    3
## 21            North Carolina  United States    3
## 22                    Oregon  United States    3
## 23                   Ontario         Canada    2
## 24                   Alabama  United States    2
## 25                     Idaho  United States    2
## 26                 Minnesota  United States    2
## 27              Mississippi  United States    2
## 28                    Nevada  United States    2
## 29                      Ohio  United States    2
## 30              Pennsylvania  United States    2
## 31                Washington  United States    2
## 32           Qarku i Tiranës        Albania    1
## 33           Khulna Division     Bangladesh    1
## 34                     Bahia         Brazil    1
## 35                   Atacama          Chile    1
## 36 Provence-Alpes-Côte d'Azur         France    1
## 37    Departamento de Olancho       Honduras    1
## 38             Andhra Pradesh          India    1
## 39                  Karnataka          India    1
## 40                   Sardegna          Italy    1
## 41                    England United Kingdom    1
## 42                    Arizona  United States    1
## 43                       Iowa  United States    1
## 44                  Louisiana  United States    1
## 45                      Maine  United States    1
## 46                   Nebraska  United States    1
## 47                   Oklahoma  United States    1
## 48             South Carolina  United States    1
## 49               South Dakota  United States    1
## 50                  Tennessee  United States    1
```

```
table(d_respondents$country) %>%
       as.data.frame() %>%
       arrange(desc(Freq))
```

```
##             Var1 Freq
## 1  United States  225
## 2          India  115
## 3         Canada    2
## 4        Albania    1
## 5     Bangladesh    1
## 6         Brazil    1
## 7          Chile    1
## 8         France    1
## 9       Honduras    1
```

```
## 10          Italy    1
## 11 United Kingdom    1
```

```
table(d_respondents$Total_Score) %>%
  as.data.frame() %>%
  arrange(desc(Var1))
```

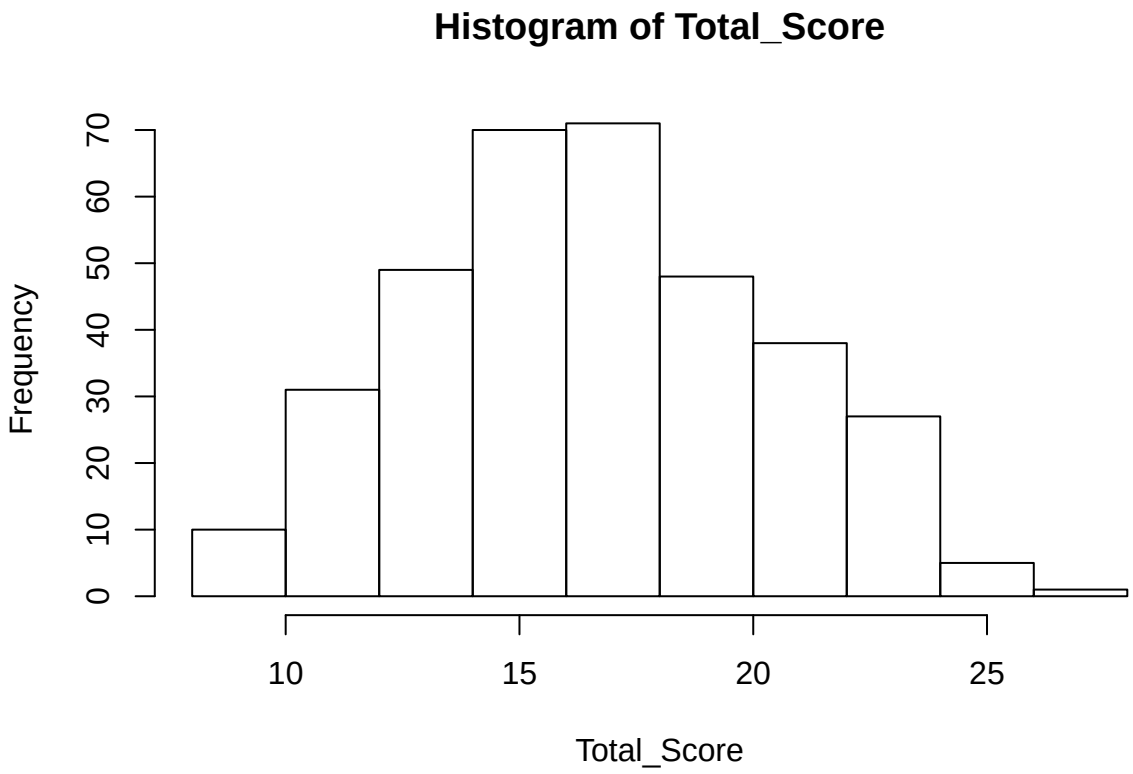
```
##    Var1 Freq
## 1    27    1
## 2    26    1
## 3    25    4
## 4    24   12
## 5    23   15
## 6    22   16
## 7    21   22
## 8    20   27
## 9    19   21
## 10   18   31
## 11   17   40
## 12   16   40
## 13   15   30
## 14   14   30
## 15   13   19
## 16   12   18
## 17   11   13
## 18   10    6
## 19    9    3
## 20    8    1
```

```
d_respondents %>%
  group_by(Assignment_Group) %>%
  summarise(mean = mean(Total_Score),
            count = n(),
            time_duration = mean(`Duration (in seconds)`))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 4
##   Assignment_Group  mean count time_duration
##   <chr>            <dbl> <int>         <dbl>
## 1 Control           16.7    69          638.
## 2 Medical Feedback  17.8    70          656.
## 3 Negative Images   16.5    72          783
## 4 Positive Images   17.3    70          505.
## 5 Self-Reflect      17.2    69          612.
```

```
#d_respondents[ , .(count = .N, avg = mean(Total_Score)), by=Assignment_Group] #same thing

d_respondents[ , hist(Total_Score)]
```




```
## $breaks
##  [1]  8 10 12 14 16 18 20 22 24 26 28
##
## $counts
##  [1] 10 31 49 70 71 48 38 27  5  1
##
## $density
##  [1] 0.014285714 0.044285714 0.070000000 0.100000000 0.101428571 0.068571429
##  [7] 0.054285714 0.038571429 0.007142857 0.001428571
##
## $mids
##  [1]  9 11 13 15 17 19 21 23 25 27
##
## $xname
## [1] "Total_Score"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
tapply(d_respondents$Total_Score, d_respondents$Assignment_Group, summary)
```

```
## $Control
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     8.0    14.0    16.0    16.7    19.0    24.0
##
## $`Medical Feedback`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   10.00   16.00   17.50   17.79   20.00   24.00
##
## $`Negative Images`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   13.00   16.00   16.51   19.25   25.00
##
## $`Positive Images`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   15.00   17.00   17.31   20.00   27.00
##
## $`Self-Reflect`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   14.00   17.00   17.25   20.00   25.00
```

```
tapply(d_respondents$Total_Score, d_respondents$Assignment_Group, sd)
```

```
##          Control Medical Feedback  Negative Images  Positive Images
##         3.659413         3.278798         3.996453         3.816603
##     Self-Reflect
##         3.882108
```

```
d_respondents[ , sd(Total_Score)]
```

```
## [1] 3.743141
```

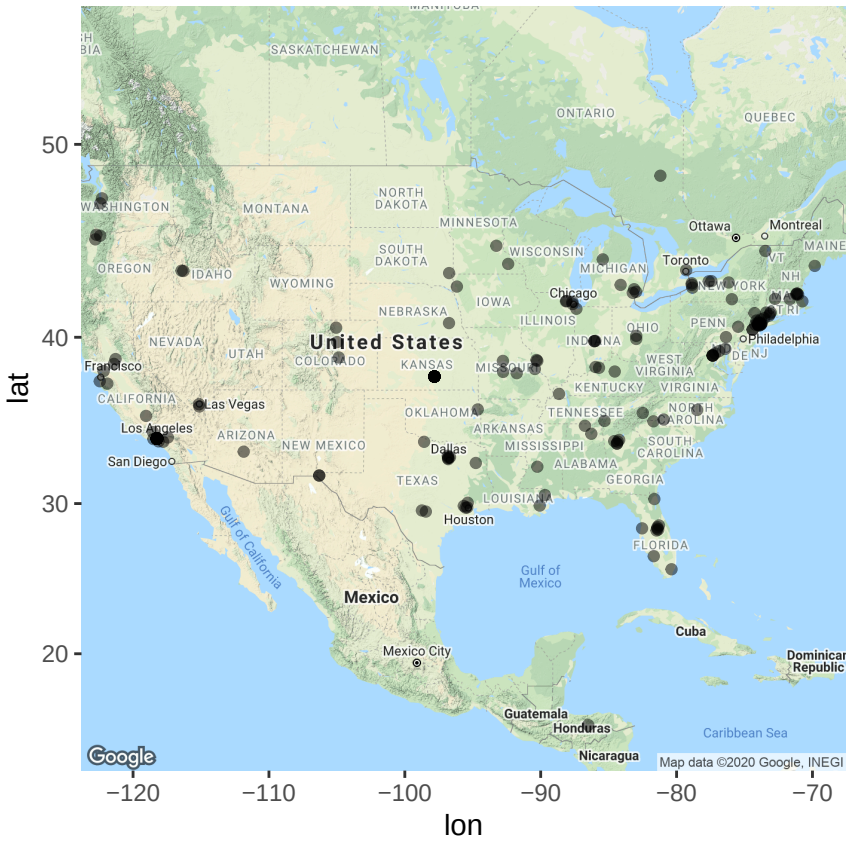
```
library(ggmap)
?register_google
register_google(key = "AIzaSyCTk2a5vIEqcvgz9KmQmItoNF7J8_hiMMk")
#ggmap_show_api_key()

us_map<-get_map(location='united states', zoom=4, maptype = "terrain",
             source='google',color='color')
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=united%20states&zoom=4&size=640x640&s
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=united+states&key=xxx
```

```
ggmap(us_map) + geom_point(x=d_respondents$LocationLongitude, y = d_respondents$LocationLatitude, show_
```

```
## Warning: `show_guide` has been deprecated. Please use `show.legend` instead.
```



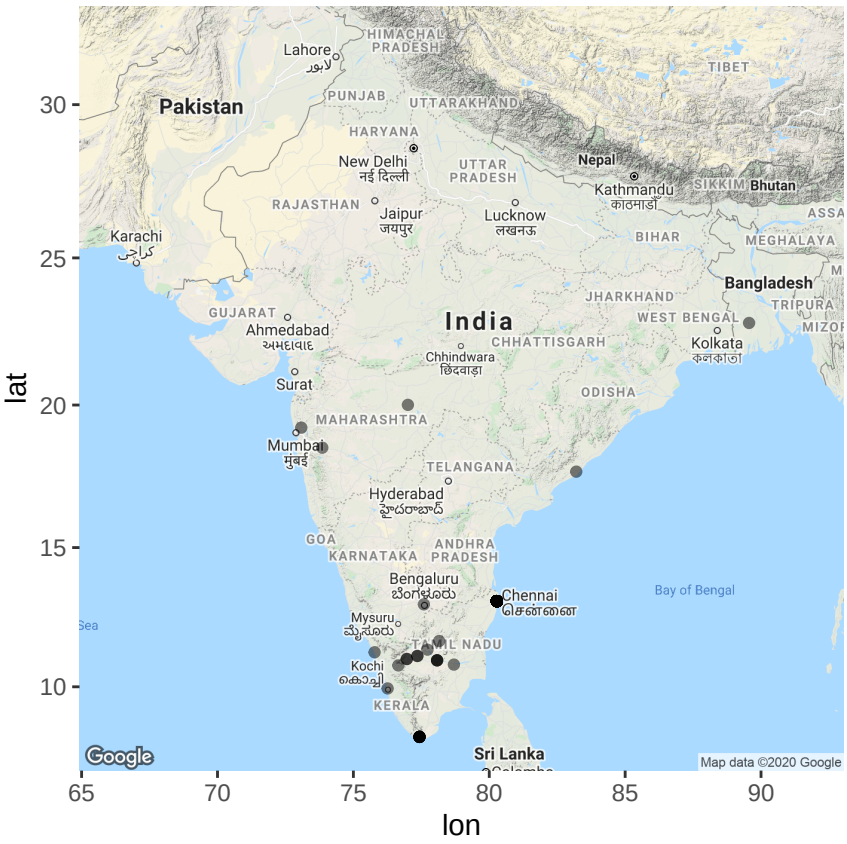

```
india_map<-get_map(location='india', zoom=5, maptype = "terrain",
             source='google',color='color')
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=india&zoom=5&size=640x640&scale=2&map
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=india&key=xxx
```

```
ggmap(india_map) + geom_point(x=d_respondents$LocationLongitude, y = d_respondents$LocationLatitude, sh
```

```
## Warning: `show_guide` has been deprecated. Please use `show.legend` instead.
```



Randomization Check

```
#http://www.sthda.com/english/wiki/chi-square-goodness-of-fit-test-in-r

respondent_counts <- d_respondents[ , .(.N), keyby=Assignment_Group][,2]

respondent_counts_chisq_test <- chisq.test(respondent_counts, p=c(1/5, 1/5, 1/5, 1/5, 1/5))

respondent_counts_chisq_test
```

```
##
##  Chi-squared test for given probabilities
##
## data:  respondent_counts
## X-squared = 0.085714, df = 4, p-value = 0.9991
```

```
#p-value = 0.9991, which is greater than significance level of 0.05.
#We can conclude that the observed proportions are not significantly different from the expected propor
```

Covariate Balance Check

```
d_respondents %>%
  group_by(Assignment_Group) %>%
  summarise(num_respondents = n(),
            pre_treatment_avg = mean(TaskPhase1_Score),
            taskphase2_avg = mean(TaskPhase2_Score),
```

```
            taskphase3_avg = mean(TaskPhase3_Score))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 5
##   Assignment_Group num_respondents pre_treatment_a~ taskphase2_avg
##   <chr>                      <int>            <dbl>          <dbl>
## 1 Control                       69            0.607          0.461
## 2 Medical Feedback              70            0.634          0.523
## 3 Negative Images               72            0.578          0.494
## 4 Positive Images               70            0.614          0.514
## 5 Self-Reflect                  69            0.599          0.526
## # ... with 1 more variable: taskphase3_avg <dbl>
```

```
#check balance between age-range, education, age
d_respondents[ , table(Assignment_Group, Gender)]
```

```
##                   Gender
## Assignment_Group   Female Male
##   Control              27   42
##   Medical Feedback     29   41
##   Negative Images      30   42
##   Positive Images      29   41
##   Self-Reflect         28   41
```

```
chisq.test(d_respondents[ , table(Assignment_Group, Gender)])
```

```
##
##  Pearson's Chi-squared test
##
## data:  d_respondents[, table(Assignment_Group, Gender)]
## X-squared = 0.12578, df = 4, p-value = 0.9981
```

```
d_respondents[ , table(Assignment_Group, Age_Range)]
```

```
##                   Age_Range
## Assignment_Group   18-24 25-34 35-44 45-54 55-64 Above 65
##   Control              5    37    11     7     9        0
##   Medical Feedback     5    38    15     4     6        2
##   Negative Images      4    38    16     9     5        0
##   Positive Images      3    31    20     5    11        0
##   Self-Reflect         3    36    10    11     8        1
```

```
# expected frequency count for each cell of the contingency table should be at least 5. Since this is n
# https://stats.stackexchange.com/questions/81483/warning-in-r-chi-squared-approximation-may-be-incorre
chisq.test(d_respondents[ , table(Assignment_Group, Age_Range)],simulate.p.value = TRUE)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  d_respondents[, table(Assignment_Group, Age_Range)]
## X-squared = 19.218, df = NA, p-value = 0.5152
```

```
d_respondents[ , table(Assignment_Group, Education_Level)]
```

```
##                   Education_Level
## Assignment_Group   Associate's degree Bachelor's degree High school
```

```
##   Control                           3                44           1
##   Medical Feedback                  0                54           1
##   Negative Images                   2                50           3
##   Positive Images                   4                45           0
##   Self-Reflect                      4                46           7
##                   Education_Level
## Assignment_Group   Master's degree and above Some high school Trade school
##   Control                                 20                0            1
##   Medical Feedback                        14                0            1
##   Negative Images                         13                1            3
##   Positive Images                         19                0            2
##   Self-Reflect                            11                0            1
```

```
chisq.test(d_respondents[ , table(Assignment_Group, Education_Level)],simulate.p.value = TRUE)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  d_respondents[, table(Assignment_Group, Education_Level)]
## X-squared = 28.7, df = NA, p-value = 0.06547
```

```
#let's consider adding age bins and education bins

d_respondents[ Age_Range == "18-24", age_bin := 1]
d_respondents[ Age_Range == "25-34", age_bin := 2]
d_respondents[ Age_Range == "35-44", age_bin := 3]
d_respondents[ Age_Range == "45-54", age_bin := 4]
d_respondents[ Age_Range == "55-64", age_bin := 5]
d_respondents[ Age_Range == "Above 65", age_bin := 6]

d_respondents[ Education_Level == "Associate's degree", edu_bin := 1]
d_respondents[ Education_Level == "Bachelor's degree", edu_bin := 2]
d_respondents[ Education_Level == "High school", edu_bin := 3]
d_respondents[ Education_Level == "Master's degree and above", edu_bin := 4]
d_respondents[ Education_Level == "Some high school", edu_bin := 5]
d_respondents[ Education_Level == "Trade school", edu_bin := 6]

d_respondents[ Assignment_Group == "Control", assign_bin := 1]
d_respondents[ Assignment_Group == "Medical Feedback", assign_bin := 2]
d_respondents[ Assignment_Group == "Negative Images", assign_bin := 3]
d_respondents[ Assignment_Group == "Positive Images", assign_bin := 4]
d_respondents[ Assignment_Group == "Self-Reflect", assign_bin := 5]

d_respondents[ , US_Dummy := ifelse(country == "United States", 1, 0)]

#head(d_respondents)
```

```
#add treatment dummy

d_respondents[ , Treatment_Dummy := ifelse(Assignment_Group != "Control", 1, 0)]
#head(d_respondents)
```

```
d_respondents[ Treatment_Dummy == 1, mean(Total_Score)] - d_respondents[ Treatment_Dummy == 0, mean(Tota
```

```
## [1] 0.5143122
```

```r
sd(d_respondents$Total_Score)
```

```
## [1] 3.743141
```

```r
d_respondents[ , lm(Total_Score ~ Education_Level)]
```

```
##
## Call:
## lm(formula = Total_Score ~ Education_Level)
##
## Coefficients:
##                                (Intercept)
##                                   19.15385
##         Education_LevelBachelor's degree
##                                   -2.30447
##                Education_LevelHigh school
##                                    0.09615
## Education_LevelMaster's degree and above
##                                   -2.07592
##          Education_LevelSome high school
##                                   -1.15385
##              Education_LevelTrade school
##                                   -0.65385
```

```r
d_respondents[ , ivreg(Total_Score ~ Education_Level | Assignment_Group)]
```

```
## Warning in ivreg.fit(X, Y, Z, weights, offset, ...): more regressors than
## instruments
```

```
##
## Call:
## ivreg(formula = Total_Score ~ Education_Level | Assignment_Group)
##
## Coefficients:
##                                (Intercept)
##                                     25.327
##         Education_LevelBachelor's degree
##                                     -4.521
##                Education_LevelHigh school
##                                    -20.380
## Education_LevelMaster's degree and above
##                                    -18.814
##          Education_LevelSome high school
##                                   -102.791
##              Education_LevelTrade school
##                                         NA
```

```r
power.t.test( delta = 1.2, sd=3.78, sig.level = 0.05, power=0.8)
```

```
##
##      Two-sample t test power calculation
##
##               n = 156.7272
##           delta = 1.2
##              sd = 3.78
##       sig.level = 0.05
```

```
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Analysis

Helper Functions

```
get_robust_se <- function(model){
  # Get robust SE for use in stargazer
  vcov <- vcovHC(model,type = "HC1")
  return(sqrt(diag(vcov)))
}
```

Task Phase 2 Analysis

```
# does any treatment have an effect on task phase 2 score?
mod_task2_a <- d_respondents[, lm(TaskPhase2_Score ~ Treatment_Dummy)]

mod_task2_b <- d_respondents[, lm(TaskPhase2_Score ~ Treatment_Dummy +
                                                     TaskPhase1_Score +
                                                     as.factor(Gender) +
                                                     as.factor(Education_Level) +
                                                     as.factor(Age_Range))]


stargazer(mod_task2_a,
          mod_task2_b,
          se = list(get_robust_se(mod_task2_a),get_robust_se(mod_task2_b)),
          omit = c("Education_Level","Age_Range"),
          add.lines = list(c('Education Fixed Effects', 'No','Yes'),
                           c('Age Fixed Effects','No','Yes')),
          type='text')
```

```
##
## ===============================================================
##                                   Dependent variable:
##                          ---------------------------------------
##                                      TaskPhase2_Score
##                                (1)                  (2)
## ---------------------------------------------------------------
## Treatment_Dummy               0.053**              0.051**
##                               (0.022)              (0.022)
##
## TaskPhase1_Score                                   0.240***
##                                                    (0.047)
##
## as.factor(Gender)Male                              -0.010
##                                                    (0.017)
##
## Constant                      0.461***             0.281***
##                               (0.019)              (0.072)
##
```

```
## ------------------------------------------------------------------------
## Education Fixed Effects          No                    Yes
## Age Fixed Effects                No                    Yes
## Observations                     350                   350
## R2                              0.017                 0.117
## Adjusted R2                     0.014                 0.083
## Residual Std. Error       0.163 (df = 348)      0.158 (df = 336)
## F Statistic             5.911** (df = 1; 348) 3.433*** (df = 13; 336)
## ========================================================================
## Note:                                        *p<0.1; **p<0.05; ***p<0.01
```

```
#does the specific treatment group have an effect on task phase 2 score?
mod_task2_c <- d_respondents[, lm(TaskPhase2_Score ~ as.factor(Assignment_Group))]

mod_task2_d <- d_respondents[, lm(TaskPhase2_Score ~ as.factor(Assignment_Group) +
                                                     TaskPhase1_Score +
                                                     as.factor(Gender) +
                                                     as.factor(Education_Level) +
                                                     as.factor(Age_Range))]

# Do you think that there are features of the data that might systematically predict that people will r
# TODO update this heterogeneity issue. I'm not quite sure this applies because they're both considered
# mod5 <- d_respondents[, lm(TaskPhase2_Score ~ Treatment_Dummy + as.factor(assign_bin) +
#                                Treatment_Dummy * as.factor(assign_bin))]
stargazer(mod_task2_c,
          mod_task2_d,
          se = list(get_robust_se(mod_task2_c),get_robust_se(mod_task2_d)),
          omit = c("Education_Level","Age_Range"),
          add.lines = list(c('Education Fixed Effects', 'No','Yes'),
                           c('Age Fixed Effects','No','Yes')),
          type='text')
```

```
##
## ===============================================================================================
##                                                      Dependent variable:
##                                                  -----------------------------------------
##                                                            TaskPhase2_Score
##                                                        (1)                    (2)
## -----------------------------------------------------------------------------------------------
## as.factor(Assignment_Group)Medical Feedback          0.062**                0.055*
##                                                     (0.027)                (0.029)
##
## as.factor(Assignment_Group)Negative Images           0.034                  0.039
##                                                     (0.027)                (0.027)
##
## as.factor(Assignment_Group)Positive Images           0.053*                 0.050*
##                                                     (0.029)                (0.027)
##
## as.factor(Assignment_Group)Self-Reflect              0.065**                0.058**
##                                                     (0.028)                (0.029)
##
## TaskPhase1_Score                                                           0.238***
##                                                                            (0.048)
##
## as.factor(Gender)Male                                                      -0.010
```

```
##                                                                                (0.017)
##
## Constant                                                    0.461***           0.282***
##                                                              (0.019)            (0.073)
##
## ------------------------------------------------------------------------------------------
## Education Fixed Effects                                        No                 Yes
## Age Fixed Effects                                              No                 Yes
## Observations                                                   350                350
## R2                                                            0.021              0.119
## Adjusted R2                                                   0.010              0.076
## Residual Std. Error                                  0.164 (df = 345)   0.158 (df = 333)
## F Statistic                                         1.874 (df = 4; 345) 2.805*** (df = 16; 333)
## ==========================================================================================
## Note:                                                        *p<0.1; **p<0.05; ***p<0.01
```

Task Phase 3 Analysis

```
# test final task and any treatment
mod_task3_a <- d_respondents[, lm(TaskPhase3_Score ~ Treatment_Dummy)]
mod_task3_b <- d_respondents[, lm(TaskPhase3_Score ~ Treatment_Dummy +
                                                     TaskPhase1_Score +
                                                     as.factor(Gender) +
                                                     as.factor(Education_Level) +
                                                     as.factor(Age_Range))]

stargazer(mod_task3_a,
          mod_task3_b,
          se = list(get_robust_se(mod_task3_a),get_robust_se(mod_task3_b)),
          omit = c("Education_Level","Age_Range"),
          add.lines = list(c('Education Fixed Effects', 'No','Yes'),
                           c('Age Fixed Effects','No','Yes')),
          type='text')
```

```
##
## ========================================================================
##                                      Dependent variable:
##                          -----------------------------------------------
##                                        TaskPhase3_Score
##                                  (1)                    (2)
## ------------------------------------------------------------------------
## Treatment_Dummy                0.004                  0.002
##                               (0.019)                (0.019)
##
## TaskPhase1_Score                                     0.161***
##                                                       (0.047)
##
## as.factor(Gender)Male                                 -0.004
##                                                       (0.017)
##
## Constant                      0.538***               0.515***
##                               (0.017)                (0.064)
##
## ------------------------------------------------------------------------
```

```
## Education Fixed Effects         No                    Yes
## Age Fixed Effects               No                    Yes
## Observations                    350                   350
## R2                            0.0001                0.084
## Adjusted R2                   -0.003                0.049
## Residual Std. Error      0.160 (df = 348)      0.155 (df = 336)
## F Statistic             0.034 (df = 1; 348) 2.384*** (df = 13; 336)
## ========================================================================
## Note:                                      *p<0.1; **p<0.05; ***p<0.01
```

```
# test final task and specific treatment
mod_task3_c <- d_respondents[, lm(TaskPhase3_Score ~ as.factor(Assignment_Group))]
mod_task3_d <- d_respondents[, lm(TaskPhase3_Score ~ as.factor(Assignment_Group) +
                                                     TaskPhase1_Score +
                                                     as.factor(Gender) +
                                                     as.factor(Education_Level) +
                                                     as.factor(Age_Range))]

stargazer(mod_task3_c,
          mod_task3_d,
          se = list(get_robust_se(mod_task3_c),get_robust_se(mod_task3_d)),
          omit = c("Education_Level","Age_Range"),
          add.lines = list(c('Education Fixed Effects', 'No','Yes'),
                           c('Age Fixed Effects','No','Yes')),
          type='text')
```

```
##
## ============================================================================================
##                                                    Dependent variable:
##                                                ----------------------------------------
##                                                          TaskPhase3_Score
##                                                      (1)                  (2)
## --------------------------------------------------------------------------------------------
## as.factor(Assignment_Group)Medical Feedback          0.022                0.011
##                                                     (0.026)              (0.026)
##
## as.factor(Assignment_Group)Negative Images          -0.015               -0.011
##                                                     (0.027)              (0.026)
##
## as.factor(Assignment_Group)Positive Images          -0.001                0.004
##                                                     (0.025)              (0.025)
##
## as.factor(Assignment_Group)Self-Reflect              0.010                0.005
##                                                     (0.026)              (0.026)
##
## TaskPhase1_Score                                                        0.157***
##                                                                         (0.047)
##
## as.factor(Gender)Male                                                   -0.004
##                                                                         (0.017)
##
## Constant                                           0.538***             0.518***
##                                                     (0.017)              (0.064)
##
## --------------------------------------------------------------------------------------------
```

```
## Education Fixed Effects                          No                  Yes
## Age Fixed Effects                                No                  Yes
## Observations                                    350                 350
## R2                                            0.006               0.087
## Adjusted R2                                  -0.005               0.043
## Residual Std. Error                   0.160 (df = 345)    0.156 (df = 333)
## F Statistic                          0.545 (df = 4; 345) 1.971** (df = 16; 333)
## ================================================================================
## Note:                                                *p<0.1; **p<0.05; ***p<0.01
```

Playground

```
### linear model playground
d_test <- d_respondents[,c("Assignment_Group","TaskPhase1_Score","TaskPhase2_Score","TaskPhase3_Score",

#does treatment have an effect on total score?
mod_test1 <- d_test[ , lm(TaskPhase2_Score ~ TaskPhase1_Score + Treatment_Dummy)]


mod_test2 <- d_test[, lm(TaskPhase2_Score ~ TaskPhase1_Score + Treatment_Dummy + (TaskPhase1_Score * Tr

#does treatment and pretreatment score have an effect on total score?


###
# seems that if i ad in TaskPhase1 to the linear model, the RSEs disappear...
mod_test3 <- d_test[, lm(TaskPhase2_Score ~  Treatment_Dummy  +
                                TaskPhase1_Score +
                                as.factor(Education_Level) +
                                as.factor(Gender) +
                                as.factor(Age_Range)
                         )]

coeftest(mod_test3, vcov = vcovHC(mod_test3,"HC1"))
```

```
##
## t test of coefficients:
##
##                                                  Estimate Std. Error
## (Intercept)                                     0.2810773  0.0721331
## Treatment_Dummy                                 0.0507118  0.0221728
## TaskPhase1_Score                                0.2402704  0.0468151
## as.factor(Education_Level)Bachelor's degree    -0.0068270  0.0485557
## as.factor(Education_Level)High school           0.0406800  0.0561886
## as.factor(Education_Level)Master's degree and above -0.0169778  0.0512819
## as.factor(Education_Level)Some high school     -0.1206457  0.0510769
## as.factor(Education_Level)Trade school          0.0286671  0.0692626
## as.factor(Gender)Male                          -0.0099516  0.0173475
## as.factor(Age_Range)25-34                       0.0446944  0.0376821
## as.factor(Age_Range)35-44                       0.0419834  0.0395245
## as.factor(Age_Range)45-54                       0.0697483  0.0417760
## as.factor(Age_Range)55-64                       0.0803500  0.0425233
## as.factor(Age_Range)Above 65                    0.1257451  0.0517169
##                                                 t value  Pr(>|t|)
```

```
## (Intercept)                                          3.8966 0.0001177 ***
## Treatment_Dummy                                      2.2871 0.0228105 *
## TaskPhase1_Score                                     5.1323 4.852e-07 ***
## as.factor(Education_Level)Bachelor's degree         -0.1406 0.8882698
## as.factor(Education_Level)High school                0.7240 0.4695746
## as.factor(Education_Level)Master's degree and above -0.3311 0.7407994
## as.factor(Education_Level)Some high school          -2.3620 0.0187444 *
## as.factor(Education_Level)Trade school               0.4139 0.6792187
## as.factor(Gender)Male                               -0.5737 0.5665811
## as.factor(Age_Range)25-34                            1.1861 0.2364245
## as.factor(Age_Range)35-44                            1.0622 0.2889014
## as.factor(Age_Range)45-54                            1.6696 0.0959341 .
## as.factor(Age_Range)55-64                            1.8896 0.0596782 .
## as.factor(Age_Range)Above 65                         2.4314 0.0155623 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(d_respondents$TaskPhase1_Score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2000  0.5000  0.6000  0.6063  0.7000  1.0000
```

```
stargazer(mod_test1,
          mod_test2,
          mod_test3,
          se = list(get_robust_se(mod_test1),get_robust_se(mod_test2), get_robust_se(mod_test3)),
          type='text')
```

```
##
## ================================================================================================
##                                                                        Dependent variable:
##                                                                -----------------------------------
##                                                                         TaskPhase2_Score
##                                                                    (1)                  (2)
## ------------------------------------------------------------------------------------------------
## TaskPhase1_Score                                                0.249***              0.153
##                                                                  (0.044)              (0.095)
##
## as.factor(Education_Level)Bachelor's degree
##
##
## as.factor(Education_Level)High school
##
##
## as.factor(Education_Level)Master's degree and above
##
##
## as.factor(Education_Level)Some high school
##
##
## as.factor(Education_Level)Trade school
##
##
## as.factor(Gender)Male
##
```

```
##
## as.factor(Age_Range)25-34
##
##
## as.factor(Age_Range)35-44
##
##
## as.factor(Age_Range)45-54
##
##
## as.factor(Age_Range)55-64
##
##
## as.factor(Age_Range)Above 65
##
##
## Treatment_Dummy                                          0.054**                -0.019
##                                                          (0.021)                (0.065)
##
## TaskPhase1_Score:Treatment_Dummy                                                 0.120
##                                                                                 (0.107)
##
## Constant                                                 0.310***               0.368***
##                                                          (0.032)                (0.057)
##
## ---------------------------------------------------------------------------------------------------------
## Observations                                               350                    350
## R2                                                         0.098                  0.101
## Adjusted R2                                                0.092                  0.093
## Residual Std. Error                                  0.157 (df = 347)       0.157 (df = 346)
## F Statistic                                       18.782*** (df = 2; 347) 12.917*** (df = 3; 346) 3
## =========================================================================================================
## Note:                                                                                         *p<0
```

```
# use Robust SE
mod_test2 <- d_respondents[, lm(TaskPhase2_Score ~ Treatment_Dummy  + as.factor(Education_Level) + (Trea
mod_test2$vcovHC_ <- vcovHC(mod_test2)
coeftest(mod_test2, vcov = mod_test2$vcovHC_)
```

```
##
## t test of coefficients:
##
##                                                            Estimate
## (Intercept)                                               0.5333333
## Treatment_Dummy                                           0.0066667
## as.factor(Education_Level)Bachelor's degree              -0.0742424
## as.factor(Education_Level)High school                    -0.0333333
## as.factor(Education_Level)Master's degree and above      -0.0733333
## as.factor(Education_Level)Some high school               -0.1400000
## as.factor(Education_Level)Trade school                   -0.2333333
## Treatment_Dummy:as.factor(Education_Level)Bachelor's degree  0.0414219
## Treatment_Dummy:as.factor(Education_Level)High school        0.0751515
## Treatment_Dummy:as.factor(Education_Level)Master's degree and above  0.0438596
## Treatment_Dummy:as.factor(Education_Level)Trade school       0.3076190
##                                                           Std. Error
```

```
## (Intercept)                                                          NA
## Treatment_Dummy                                                      NA
## as.factor(Education_Level)Bachelor's degree                          NA
## as.factor(Education_Level)High school                                NA
## as.factor(Education_Level)Master's degree and above                  NA
## as.factor(Education_Level)Some high school                           NA
## as.factor(Education_Level)Trade school                               NA
## Treatment_Dummy:as.factor(Education_Level)Bachelor's degree          NA
## Treatment_Dummy:as.factor(Education_Level)High school                NA
## Treatment_Dummy:as.factor(Education_Level)Master's degree and above  NA
## Treatment_Dummy:as.factor(Education_Level)Trade school               NA
##                                                                  t value
## (Intercept)                                                           NA
## Treatment_Dummy                                                       NA
## as.factor(Education_Level)Bachelor's degree                           NA
## as.factor(Education_Level)High school                                 NA
## as.factor(Education_Level)Master's degree and above                   NA
## as.factor(Education_Level)Some high school                            NA
## as.factor(Education_Level)Trade school                                NA
## Treatment_Dummy:as.factor(Education_Level)Bachelor's degree           NA
## Treatment_Dummy:as.factor(Education_Level)High school                 NA
## Treatment_Dummy:as.factor(Education_Level)Master's degree and above   NA
## Treatment_Dummy:as.factor(Education_Level)Trade school                NA
##                                                                 Pr(>|t|)
## (Intercept)                                                           NA
## Treatment_Dummy                                                       NA
## as.factor(Education_Level)Bachelor's degree                           NA
## as.factor(Education_Level)High school                                 NA
## as.factor(Education_Level)Master's degree and above                   NA
## as.factor(Education_Level)Some high school                            NA
## as.factor(Education_Level)Trade school                                NA
## Treatment_Dummy:as.factor(Education_Level)Bachelor's degree           NA
## Treatment_Dummy:as.factor(Education_Level)High school                 NA
## Treatment_Dummy:as.factor(Education_Level)Master's degree and above   NA
## Treatment_Dummy:as.factor(Education_Level)Trade school                NA
```