

Feedback MTurk Study

Dahler Battle, Guy El Khoury, Jane Hung, and Julian Tsang

15 Dec 2020

Introduction

Load Data

```
# ?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("houenumber", "street", "city", "county", "state", "zip", "country") := revu]
# #
# head(d_respondents_only)
# #
# #
# fwrite(d_respondents_only, file='datatable_clean_survey_responses_v2.dta')

d_respondents <- fread('datatable_clean_survey_responses_v2.dta')

setnames(d_respondents,
  old = c('Duration (in seconds)'),
  new = c('Survey_Duration'))
kable(t(head(d_respondents[, names(d_respondents)[!names(d_respondents) %in% c(
  "Q70_First Click", "Q70_Last Click", "Q70_Page Submit", "Q70_Click Count",
  "Q90_First Click", "Q90_Last Click", "Q90_Page Submit", "Q90_Click Count",
  "Q61_First Click", "Q61_Last Click", "Q61_Page Submit", "Q61_Click Count",
  "Q62_First Click", "Q62_Last Click", "Q62_Page Submit", "Q62_Click Count",
  "Q63_First Click", "Q63_Last Click", "Q63_Page Submit", "Q63_Click Count",
  "Q64_First Click", "Q64_Last Click", "Q64_Page Submit", "Q64_Click Count",
  "Q65_First Click", "Q65_Last Click", "Q65_Page Submit", "Q65_Click Count",
  "Q66_First Click", "Q66_Last Click", "Q66_Page Submit", "Q66_Click Count",
  "Q67_First Click", "Q67_Last Click", "Q67_Page Submit", "Q67_Click Count",
  "Q68_First Click", "Q68_Last Click", "Q68_Page Submit", "Q68_Click Count",
  "houenumber", "street", "zip")],with=FALSE),3)))
```

StartDate	2020-11-09 20:46:55	2020-11-09 20:47:33	2020-11-09 20:47:23
EndDate	2020-11-09 20:50:39	2020-11-09 20:51:24	2020-11-09 20:51:35
Status	IP Address	IP Address	IP Address
IPAddress	174.88.123.135	172.93.166.91	68.36.215.223
Progress	100	100	100
Survey_Duration	223	231	251
Finished	TRUE	TRUE	TRUE
RecordedDate	2020-11-09 20:50:39	2020-11-09 20:51:25	2020-11-09 20:51:35

ResponseId	R_VLuUQ4C82PP9HEd	R_29cCZD1XK1dpmdY	R_3lVN8EncJofnqnV
RecipientLastName	NA	NA	NA
RecipientFirstName	NA	NA	NA
RecipientEmail	NA	NA	NA
ExternalReference	NA	NA	NA
LocationLatitude	43.68	33.75	42.66
LocationLongitude	-79.29	-84.39	-83.12
DistributionChannel	anonymous	anonymous	anonymous
UserLanguage	EN	EN	EN
Amazon_Turk_ID	A4D99Y82KOLC8	A1AC47WJLNW4G7	A77K8W55MJEKX
Gender	Male	Male	Female
Q82_3_TEXT	NA	NA	NA
Age_Range	35-44	25-34	45-54
Education_Level	Trade school	Master's degree and above	Bachelor's degree
Q1	Pneumonia	Pneumonia	Pneumonia
Q2	Normal	Normal	Pneumonia
Q3	Normal	Normal	Pneumonia
Q4	Pneumonia	Pneumonia	Pneumonia
Q5	Normal	Normal	Pneumonia
Q6	Pneumonia	Normal	Pneumonia
Q7	Pneumonia	Normal	Normal
Q8	Normal	Normal	Normal
Q9	Pneumonia	Normal	Pneumonia
Q10	Pneumonia	Pneumonia	Pneumonia
Control_Q1			
Q11	Normal	Normal	Pneumonia
Q12	Normal	Normal	Normal
Q13	Pnuemonia	Pnuemonia	Pnuemonia
Q14	Pneumonia	Normal	Pneumonia
Q15	Pneumonia	Normal	Pneumonia
Q16	Normal	Pneumonia	Pneumonia
Q17	Normal	Normal	Pneumonia
Q18	Pneumonia	Pneumonia	Pneumonia
Q19	Pneumonia	Normal	Pneumonia
Q20	Normal	Pneumonia	Pneumonia
Control_Q2			
Q21	Pneumonia	Normal	Pneumonia
Q22	Pneumonia	Pneumonia	Pneumonia
Q23	Normal	Pneumonia	Normal
Q24	Pneumonia	Normal	Pneumonia
Q25	Normal	Normal	Pneumonia
Q26	Pneumonia	Normal	Pneumonia
Q27	Pneumonia	Pneumonia	Pneumonia
Q28	Normal	Normal	Pneumonia
Q29	Normal	Pneumonia	Pneumonia
Q30	Pneumonia	Normal	Pneumonia
Q36			
Self_Reflect_Q1			
Q41			
Self_Reflect_Q2			
Q38			
Q43			
Q45	NA	NA	NA

Q47	NA	NA	NA
Q46	NA	NA	NA
Q48	NA	NA	NA
Total_Score	16	12	15
Random ID	14409	58508	96075
Assignment	FL_41	FL_16	FL_41
Q1_Score	0	0	0
Q2_Score	1	1	0
Q3_Score	0	0	1
Q4_Score	1	1	1
Q5_Score	1	1	0
Q6_Score	1	0	1
Q7_Score	1	0	0
Q8_Score	1	1	1
Q9_Score	1	0	1
Q10_Score	0	0	0
Q11_Score	0	0	1
Q12_Score	1	1	1
Q13_Score	0	0	0
Q14_Score	1	0	1
Q15_Score	0	1	0
Q16_Score	1	0	0
Q17_Score	0	0	1
Q18_Score	0	0	0
Q19_Score	1	0	1
Q20_Score	1	0	0
Q21_Score	0	1	0
Q22_Score	0	0	0
Q23_Score	0	1	0
Q24_Score	0	1	0
Q25_Score	0	0	1
Q26_Score	1	0	1
Q27_Score	1	1	1
Q28_Score	0	0	1
Q29_Score	1	0	0
Q30_Score	0	1	0
Assignment_Group	Negative Images	Positive Images	Negative Images
TaskPhase1_Score	0.7	0.4	0.5
TaskPhase2_Score	0.5	0.2	0.5
TaskPhase3_Score	0.3	0.5	0.4
city	Toronto	Atlanta	Rochester Hills
county	Canada	United States	United States
state	Ontario	Georgia	Michigan
country	Canada	United States	United States

```
nrow(d_respondents)
```

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

```
#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

Helper Functions

```
create_heatmap <- function(var1, var2) {
  ### Create a heatmap for a table of frequencies between two variables ###
  df <- data.frame(table(var1,var2))

  ggplot(df,aes(x=var1,y=var2)) +
    geom_tile(aes(fill=Freq,color=Freq),show.legend=FALSE,alpha=.8) +
    geom_text(aes(label=Freq)) +
    scale_fill_continuous(high = "darkslategray4", low = "powderblue")
}

g_legend<-function(a.gplot){
  #extract legend from a ggplot object
  #https://stackoverflow.com/questions/13649473/add-a-common-legend-for-combined-ggplots
  #https://github.com/hadley/ggplot2/wiki/Share-a-legend-between-two-ggplot2-graphs
  tmp <- ggplot_gtable(ggplot_build(a.gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)}

```

#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(Survey_Duration))
```

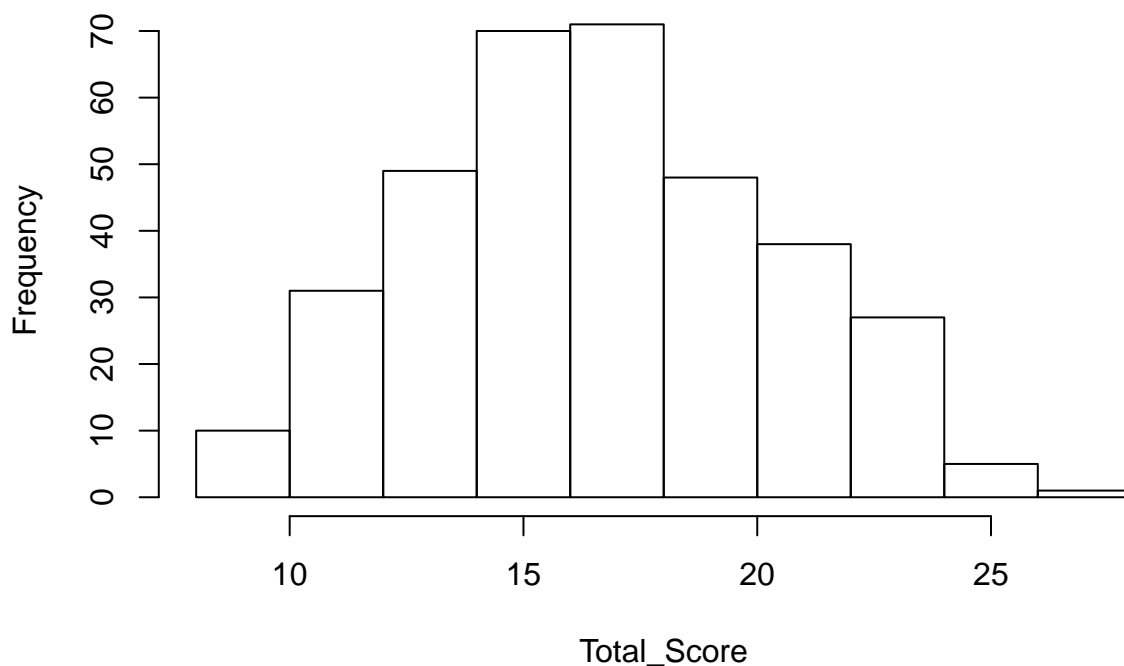
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # 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)]
```

Histogram of 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] 0.014286 0.044286 0.070000 0.100000 0.101429 0.068571 0.054286 0.038571
## [9] 0.007143 0.001429
##
## $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.
```

```
##      10.0      16.0      17.5      17.8      20.0      24.0
##
## $`Negative Images`
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      9.0      13.0      16.0      16.5      19.2      25.0
##
## $`Positive Images`
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      9.0      15.0      17.0      17.3      20.0      27.0
##
## $`Self-Reflect`
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      9.0      14.0      17.0      17.2      20.0      25.0
```

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

```
##           Control Medical Feedback  Negative Images  Positive Images
##           3.659                3.279                3.996                3.817
##      Self-Reflect
##           3.882
```

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

```
## [1] 3.743
```

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

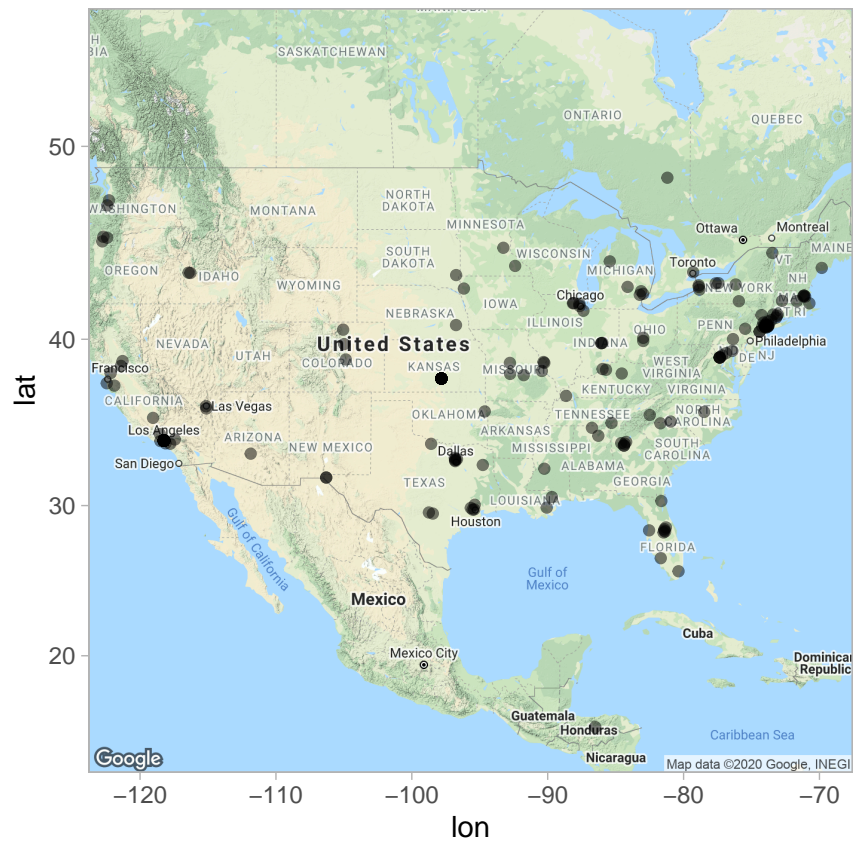
us_map<-get_map(location='united states', zoom=4, maptype = "terrain",
                 source='google',color='color')
```

```
## 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.
```

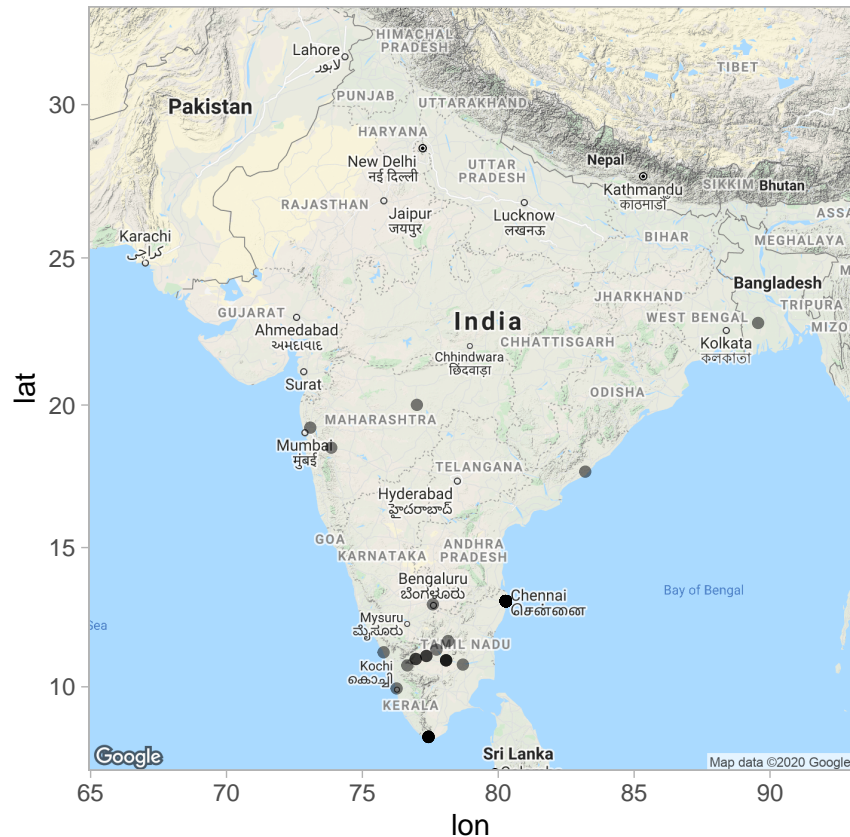
```
india_map<-get_map(location='india', zoom=5, maptype = "terrain",
  source='google',color='color')
```

```
## 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]

respondent_counts_chisq_test <- chisq.test(respondent_counts[,2], p=c(1/5, 1/5, 1/5, 1/5, 1/5))
pander(respondent_counts_chisq_test, style = 'rmarkdown')
```

Table 2: Chi-squared test for given probabilities:
respondent_counts[, 2]

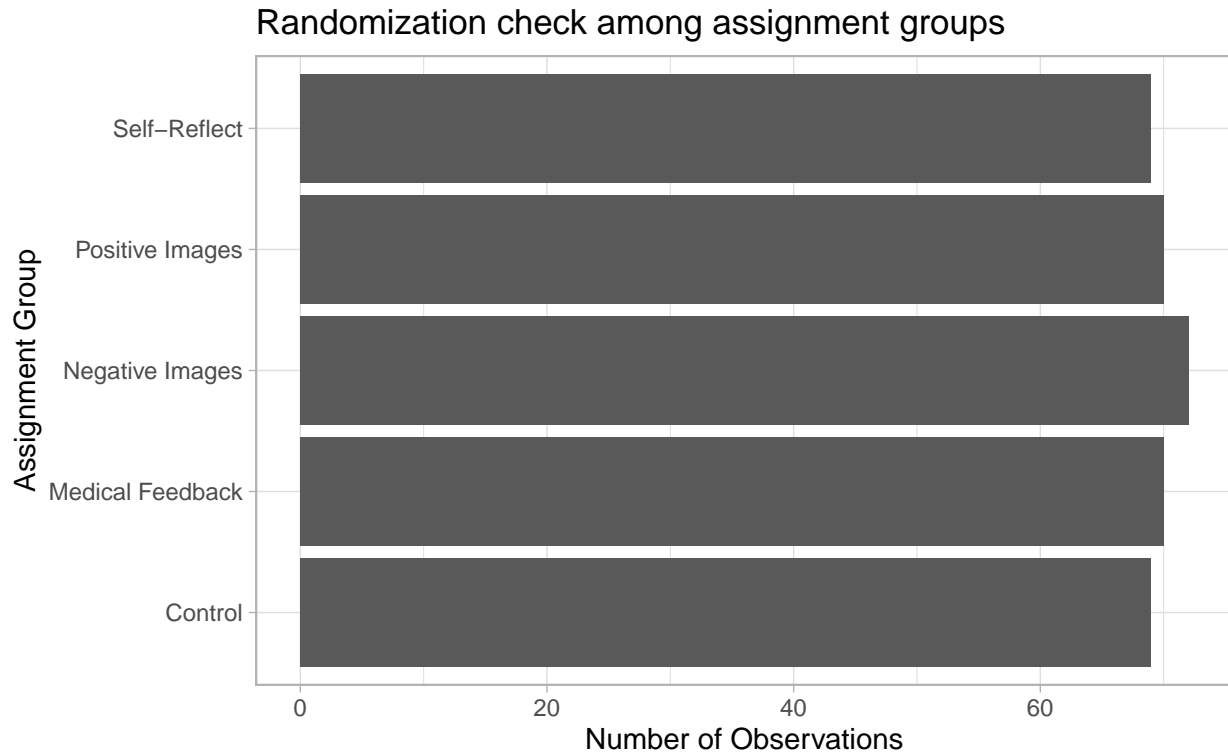
Test statistic	df	P value
0.08571	4	0.9991

```
respondent_counts %>%
  ggplot(aes(x = Assignment_Group, y=N)) +
  geom_bar(stat = 'identity') +
  coord_flip() +
  xlab('Assignment Group') +
  ylab('Number of Observations') +
  labs(title='Randomization check among assignment groups',
       caption = paste0('Assuming equal distribution among assignment groups, a chi-squared goodness of
round(respondent_counts_chisq_test$parameter,4),' degrees of \nfreedom ', 'yield
round(respondent_counts_chisq_test$p.value,4),
```

```

', suggesting that the observed proportions are not significantly different from
theme(plot.caption = element_text(hjust = 0))

```



Assuming equal distribution among assignment groups, a chi-squared goodness of fit test with 4 degrees of freedom yields $p=0.9991$, suggesting that the observed proportions are not significantly different from the expected proportions at a significance level of 0.05.

*#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 proportions*

Covariate Balance Check

#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)]

d_respondents[ , Male_Dummy := ifelse(Gender == "Male", 1, 0)]

#add treatment dummy

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

#head(d_respondents)

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>

d_respondents %>%
  group_by(Assignment_Group) %>%
  summarise(num_respondents = n(),
            avg_age_bin = mean(age_bin),
            avg_edu_bin = mean(edu_bin),
            male = mean(Male_Dummy),
            US = mean(US_Dummy))

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

## # A tibble: 5 x 6
##   Assignment_Group num_respondents avg_age_bin avg_edu_bin male    US
##   <chr>            <int>          <dbl>          <dbl> <dbl> <dbl>
## 1 Control          69            2.68            2.61 0.609 0.652
## 2 Medical Feedback 70            2.63            2.47 0.586 0.529
## 3 Negative Images  72            2.62            2.58 0.583 0.625
## 4 Positive Images  70            2.86            2.6  0.586 0.714
## 5 Self-Reflect     69            2.83            2.42 0.594 0.696

d_respondents %>%
  group_by(Assignment_Group) %>%
  summarise(num_respondents = n(),
            )

```

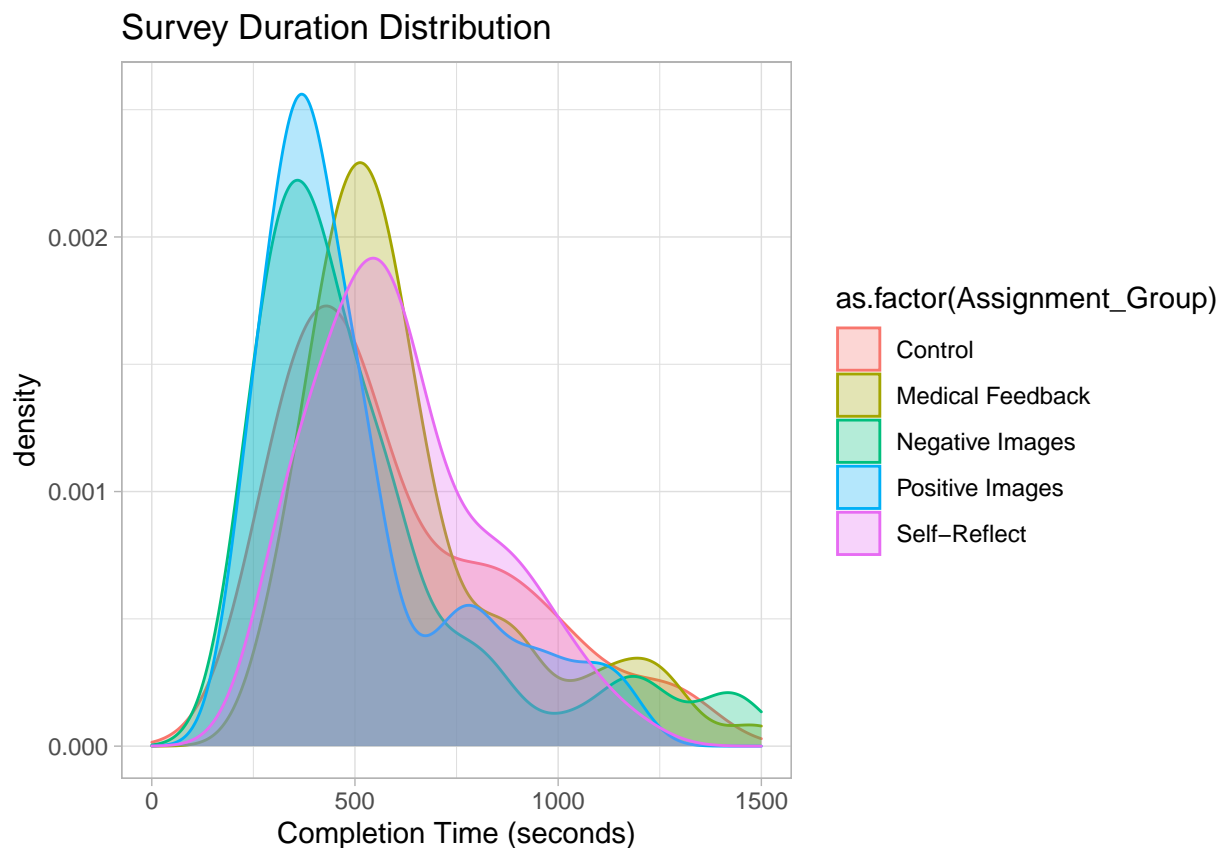
```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 5 x 2
##   Assignment_Group num_respondents
##   <chr>             <int>
## 1 Control           69
## 2 Medical Feedback  70
## 3 Negative Images   72
## 4 Positive Images   70
## 5 Self-Reflect      69
```

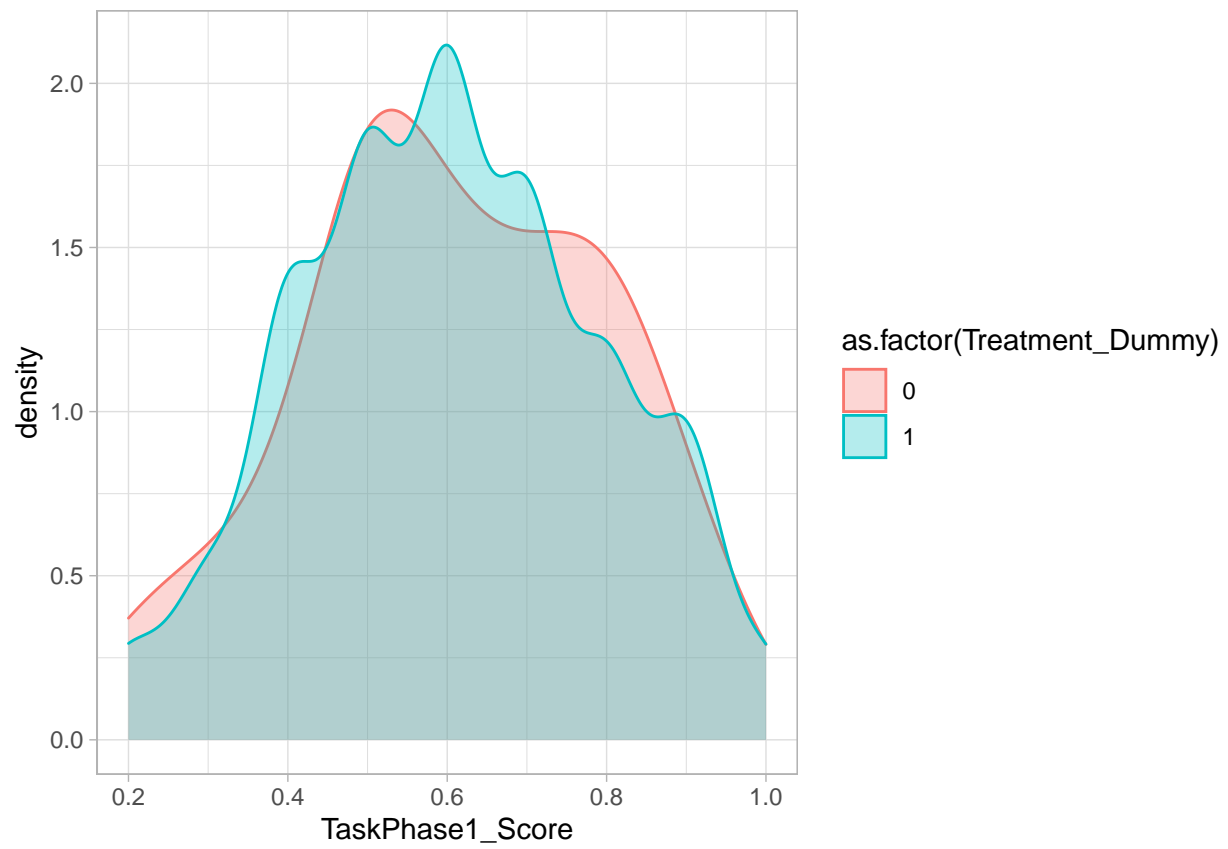
Visuals

```
#Density distribution of Survey Duration
ggplot(d_respondents, aes(x=Survey_Duration, colour=as.factor(Assignment_Group), fill = as.factor(Assignment_Group))) +
  geom_density(alpha = 0.3) +
  xlim(0, 1500) +
  xlab("Completion Time (seconds)") +
  ggtitle("Survey Duration Distribution")
```

```
## Warning: Removed 6 rows containing non-finite values (stat_density).
```

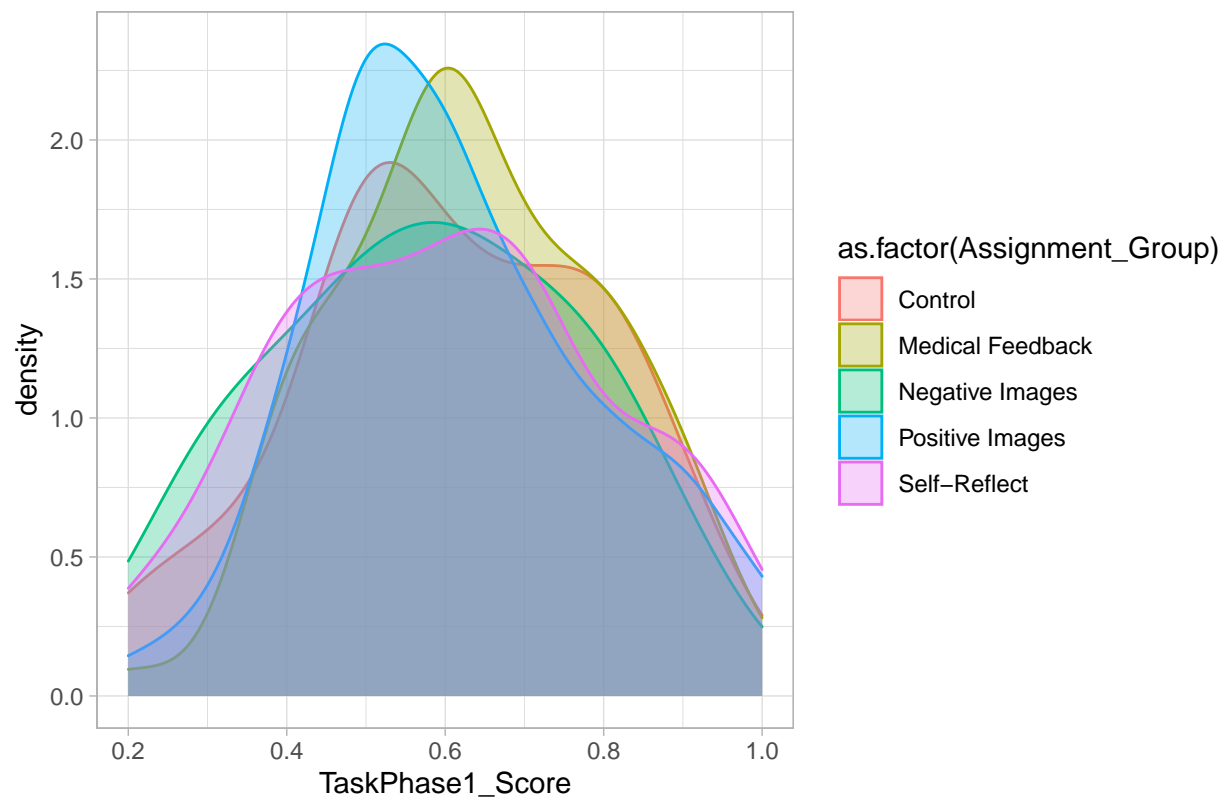


```
#Comparing pretreatment values
ggplot(d_respondents, aes(x=TaskPhase1_Score, fill = as.factor(Treatment_Dummy), colour=as.factor(Treatment_Dummy))) +
```



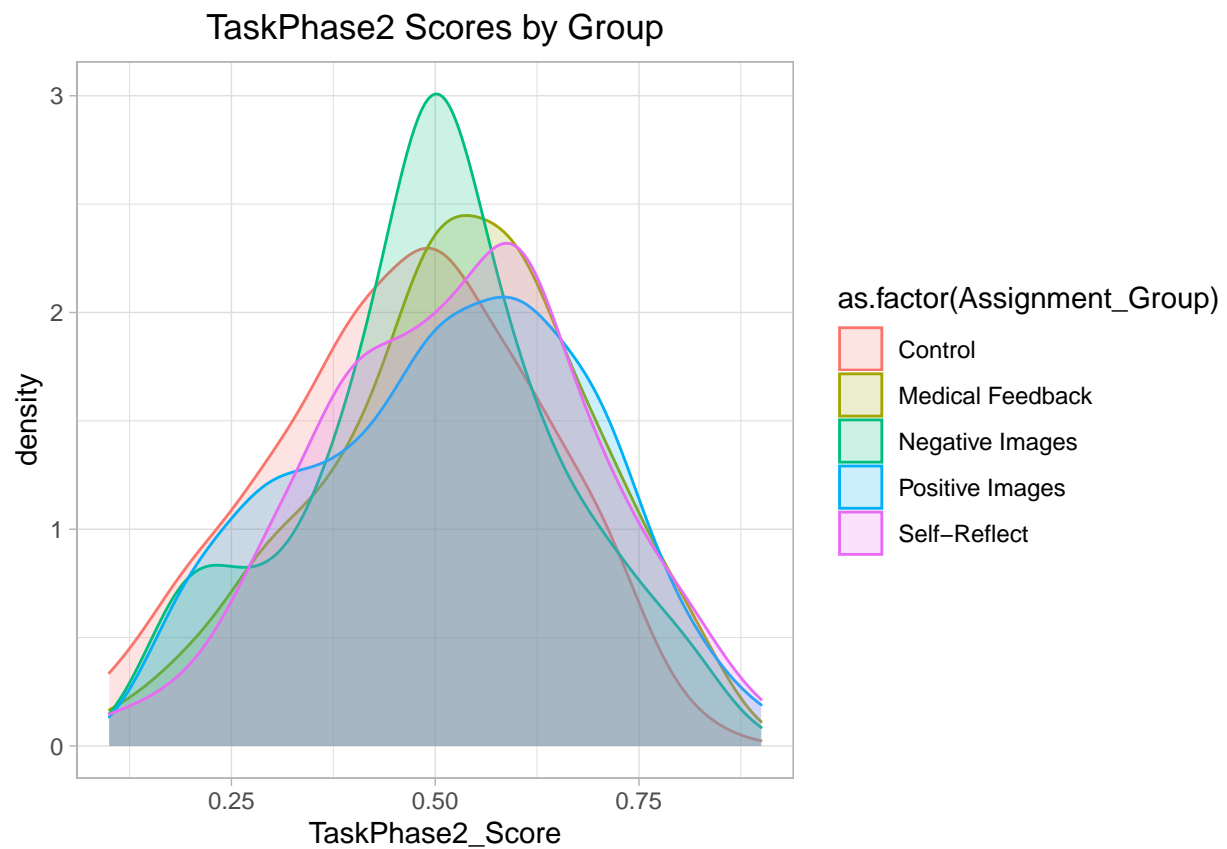
```
ggplot(d_respondents, aes(x=TaskPhase1_Score, fill = as.factor(Assignment_Group), colour=as.factor(Assi
```

PreTreatment Scores by Group



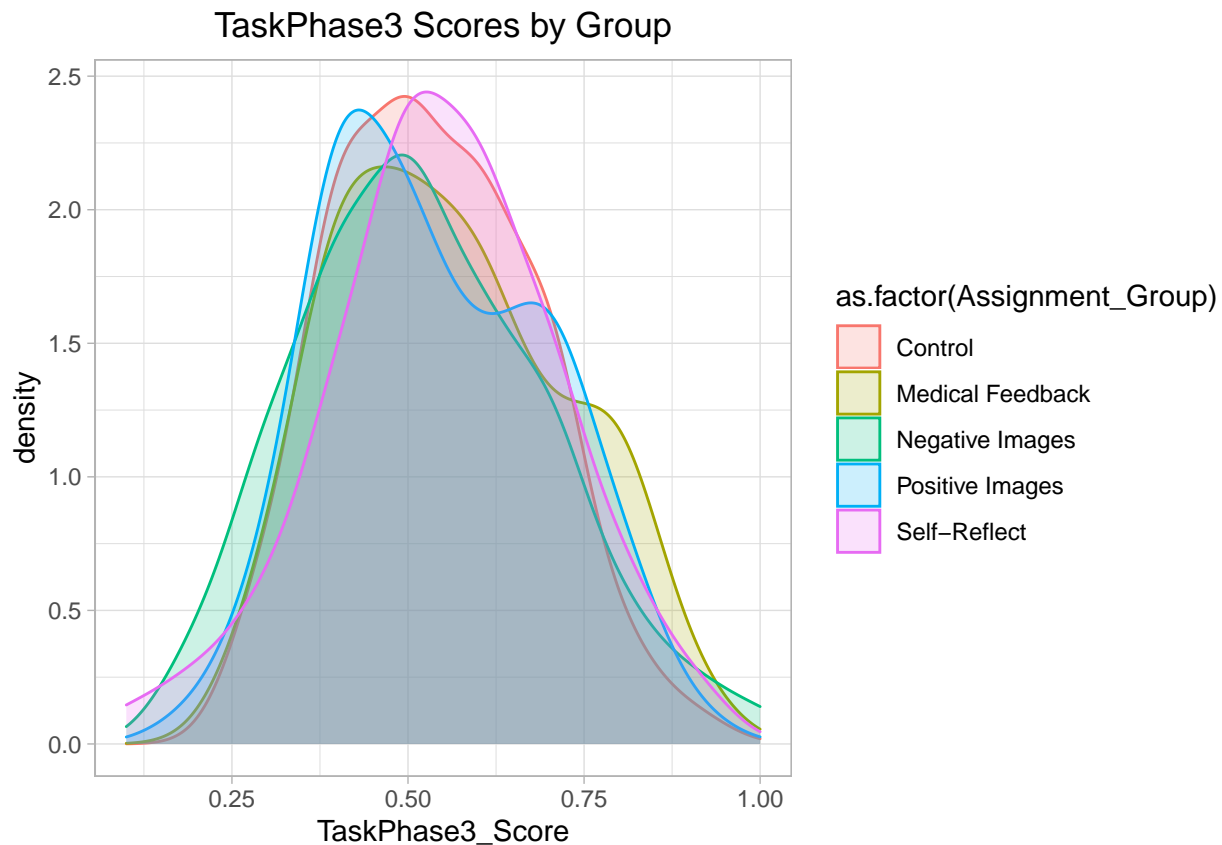
#Comparing taskphase2 values

```
ggplot(d_respondents, aes(x=TaskPhase2_Score, fill = as.factor(Assignment_Group), colour=as.factor(Assignment_Group)))
```



```
#Comparing taskphase3 values
```

```
ggplot(d_respondents, aes(x=TaskPhase3_Score, fill = as.factor(Assignment_Group), colour=as.factor(Assignment_Group)))
```

TODO add these boxplots for binary treatment/control groups

```
task2a_bp <- ggplot(d_respondents, aes(x = Treatment_Dummy, y=TaskPhase1_Score, colour=as.factor(Treatment_Dummy))) +
  geom_boxplot() +
  stat_summary(fun.y = mean, geom = "errorbar", aes(ymax = ..y.., ymin = ..y..), width = .75, linetype = "solid") +
  xlab('') +
  ylab('Task Score (%)') +
  ggtitle("Pre Treatment Scores") +
  scale_y_continuous(labels = scales::percent, limits = c(0,1)) +
  theme(axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size=10),
        legend.position = "bottom",
        legend.title = element_blank())
```

Warning: `fun.y` is deprecated. Use `fun` instead.

```
task2b_bp <- ggplot(d_respondents, aes(x = Treatment_Dummy, y=TaskPhase2_Score, colour=as.factor(Treatment_Dummy))) +
  geom_boxplot() +
  stat_summary(fun.y = mean, geom = "errorbar", aes(ymax = ..y.., ymin = ..y..), width = .75, linetype = "solid") +
  xlab('') +
  ylab('') +
  ggtitle("Task Phase 2 Scores") +
  scale_y_continuous(labels = scales::percent, limits = c(0,1)) +
  theme(axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size=10),
        legend.position = "none")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

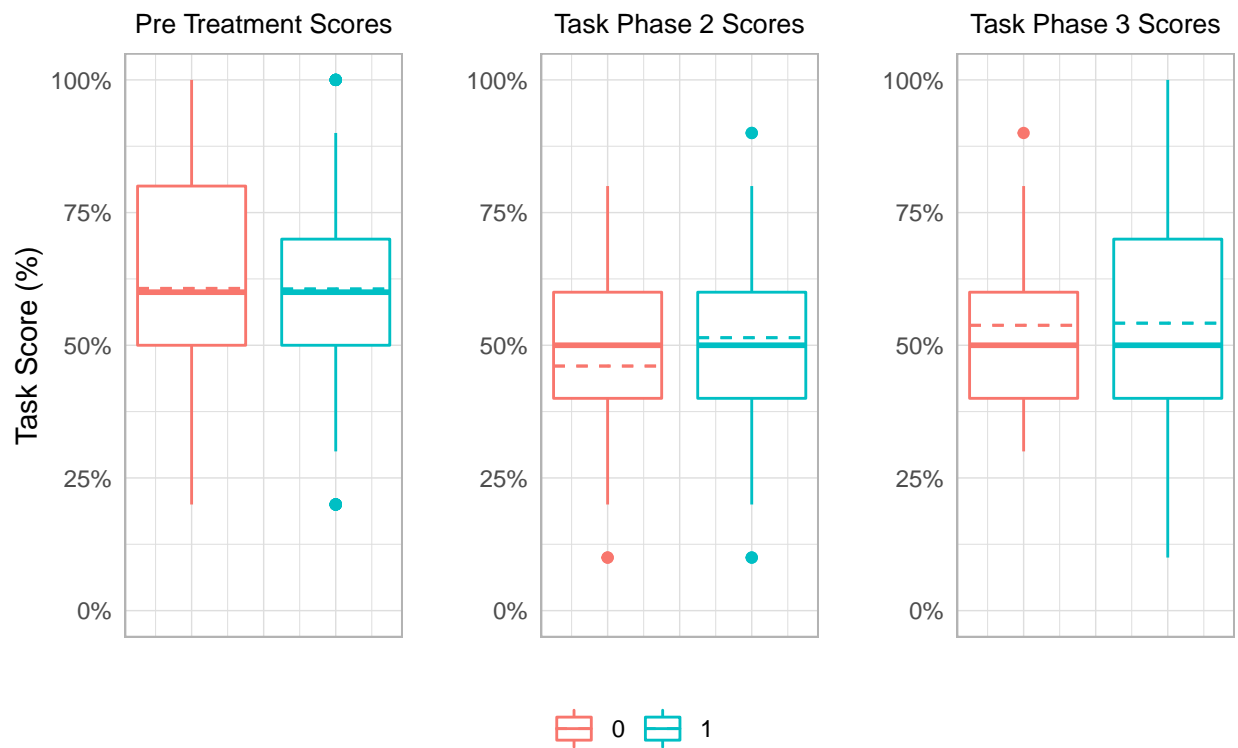
```
task2c_bp <- ggplot(d_respondents, aes(x = Treatment_Dummy, y=TaskPhase3_Score, colour=as.factor(Treatment_Dummy))) +
  geom_boxplot() +
  stat_summary(fun.y = mean, geom = "errorbar", aes(ymax = ..y.., ymin = ..y..), width = .75, linetype="dashed") +
  xlab('') +
  ylab('') +
  ggtitle("Task Phase 3 Scores") +
  scale_y_continuous(labels = scales::percent, limits = c(0,1)) +
  theme(axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size=10),
        legend.position = "none")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

```
mylegend_2<-g_legend(task2a_bp)
```

```
grid.arrange(arrangeGrob(task2a_bp + theme(legend.position="none"), task2b_bp, task2c_bp, ncol=3),
  mylegend_2,
  nrow=2,
  heights=c(10,1),
  top = textGrob("Compare task scores in different phases\n", just='right', gp=gpar(fontsize=14)))
```

Compare task scores in different phases



```
# boxplots for multiple treatment groups
```

```
task1a_bp <- ggplot(d_respondents, aes(x = Assignment_Group, y=TaskPhase1_Score, colour=as.factor(Assignment_Group))) +
  geom_boxplot() +
```

```

stat_summary(fun.y = mean, geom = "errorbar", aes(ymax = ..y.., ymin = ..y..), width = .75, linetype = "solid",
xlab('') +
ylab('Task Score (%)') +
ggtitle("Pre Treatment Scores") +
scale_y_continuous(labels = scales::percent,limits = c(0,1)) +
theme(axis.text.x = element_blank(),
axis.ticks = element_blank(),
plot.title = element_text(hjust = 0.5,size=10),
legend.position = "bottom",
legend.title = element_blank())

## Warning: `fun.y` is deprecated. Use `fun` instead.

task1b_bp <- ggplot(d_respondents, aes(x = Assignment_Group, y=TaskPhase2_Score, colour=as.factor(Assignment_Group)) +
geom_boxplot() +
stat_summary(fun.y = mean, geom = "errorbar", aes(ymax = ..y.., ymin = ..y..), width = .75, linetype = "solid",
xlab('') +
ylab('') +
ggtitle("Task Phase 2 Scores") +
scale_y_continuous(labels = scales::percent,limits = c(0,1)) +
theme(axis.text.x = element_blank(),
axis.ticks = element_blank(),
plot.title = element_text(hjust = 0.5,size=10),
legend.position = "none")

## Warning: `fun.y` is deprecated. Use `fun` instead.

task1c_bp <- ggplot(d_respondents, aes(x = Assignment_Group, y=TaskPhase3_Score, colour=as.factor(Assignment_Group)) +
geom_boxplot() +
stat_summary(fun.y = mean, geom = "errorbar", aes(ymax = ..y.., ymin = ..y..), width = .75, linetype = "solid",
xlab('') +
ylab('') +
ggtitle("Task Phase 3 Scores") +
scale_y_continuous(labels = scales::percent,limits = c(0,1)) +
theme(axis.text.x = element_blank(),
axis.ticks = element_blank(),
plot.title = element_text(hjust = 0.5,size=10),
legend.position = "none")

## Warning: `fun.y` is deprecated. Use `fun` instead.

mylegend_1<-g_legend(task1a_bp)

grid.arrange(arrangeGrob(task1a_bp + theme(legend.position="none"),task1b_bp,task1c_bp,ncol=3),
mylegend_1,
nrow=2,
heights=c(10,1),
top = textGrob("Compare task scores in different phases\n",just='right',gp=gpar(fontsize=14)))

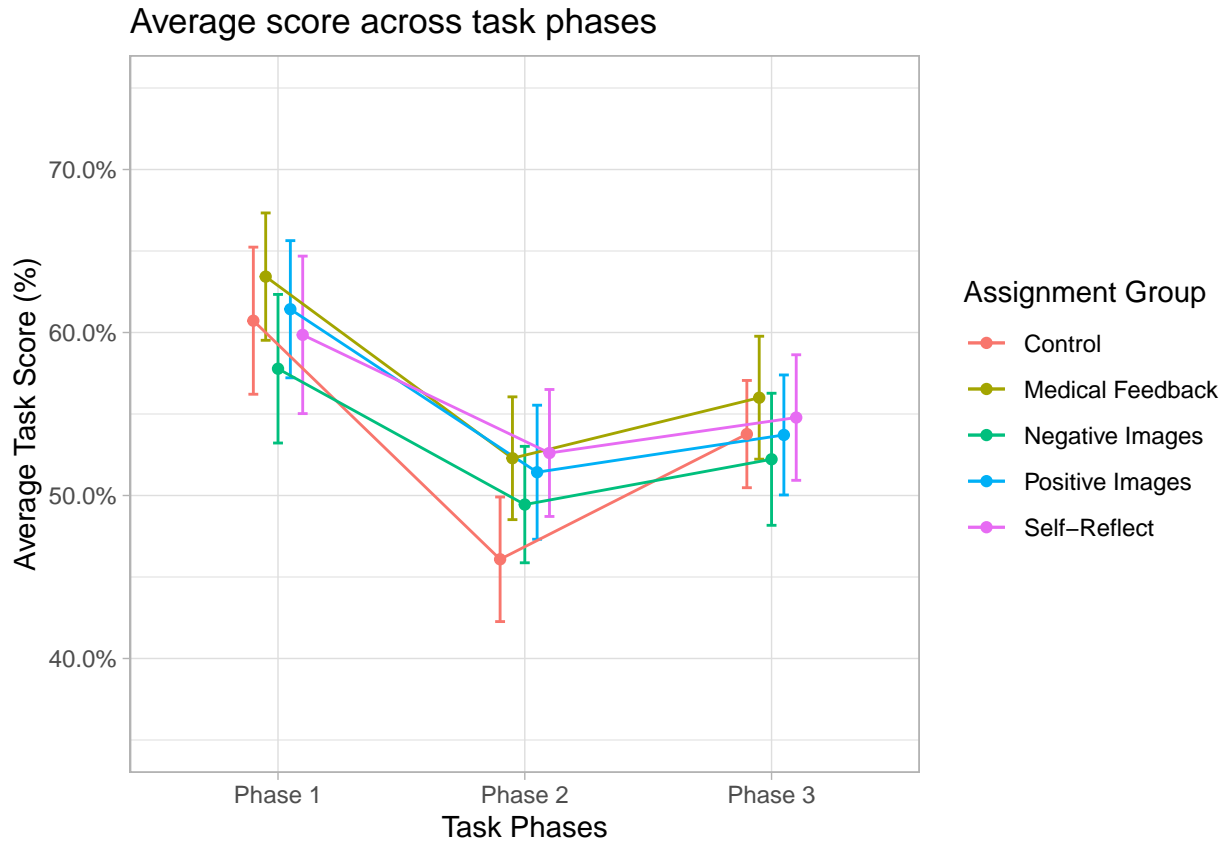
```

Compare task scores in different phases



```
# Compare score across time for all groups
# https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Confidence_Intervals/BS704_Confidence_Intervals_
# TODO finish formatting
# TODO duplicate for treatment dummy as well
summary_task_score <- (melt(d_respondents,id.vars=c('Assignment_Group'),
                           measure.vars = c('TaskPhase1_Score','TaskPhase2_Score','TaskPhase3_Score'))
  ,.('avg_score'=mean(value),'sd_score'=sd(value),'obs'=.N),keyby=.(Assignment_Group,variable))[,
  ,se:=1.96*sd_score/sqrt(obs)]

summary_task_score %>%
  ggplot( aes(x=variable, y=avg_score, group=Assignment_Group, color=Assignment_Group)) +
  geom_errorbar(aes(ymin=avg_score-1.96*sd_score/sqrt(obs), ymax=avg_score+1.96*sd_score/sqrt(obs)),
    width=.2,
    position=position_dodge(0.25)) +
  geom_line(position=position_dodge(0.25)) +
  geom_point(position=position_dodge(0.25)) +
  scale_y_continuous(labels = scales::percent,limits = c(.35,.75)) +
  scale_x_discrete(breaks=c("TaskPhase1_Score", "TaskPhase2_Score","TaskPhase3_Score"),
    labels=c("Phase 1", "Phase 2", "Phase 3")) +
  xlab('Task Phases') +
  ylab('Average Task Score (%)') +
  labs(title='Average score across task phases', color = "Assignment Group")
```



```
# TODO add this to the appendix
kable(summary_task_score)
```

Assignment_Group	variable	avg_score	sd_score	obs	se
Control	TaskPhase1_Score	0.6072	0.1912	69	0.0451
Control	TaskPhase2_Score	0.4609	0.1620	69	0.0382
Control	TaskPhase3_Score	0.5377	0.1394	69	0.0329
Medical Feedback	TaskPhase1_Score	0.6343	0.1667	70	0.0391
Medical Feedback	TaskPhase2_Score	0.5229	0.1608	70	0.0377
Medical Feedback	TaskPhase3_Score	0.5600	0.1610	70	0.0377
Negative Images	TaskPhase1_Score	0.5778	0.1973	72	0.0456
Negative Images	TaskPhase2_Score	0.4944	0.1546	72	0.0357
Negative Images	TaskPhase3_Score	0.5222	0.1754	72	0.0405
Positive Images	TaskPhase1_Score	0.6143	0.1796	70	0.0421
Positive Images	TaskPhase2_Score	0.5143	0.1755	70	0.0411
Positive Images	TaskPhase3_Score	0.5371	0.1571	70	0.0368
Self-Reflect	TaskPhase1_Score	0.5986	0.2047	69	0.0483
Self-Reflect	TaskPhase2_Score	0.5261	0.1651	69	0.0390
Self-Reflect	TaskPhase3_Score	0.5478	0.1632	69	0.0385

Gender

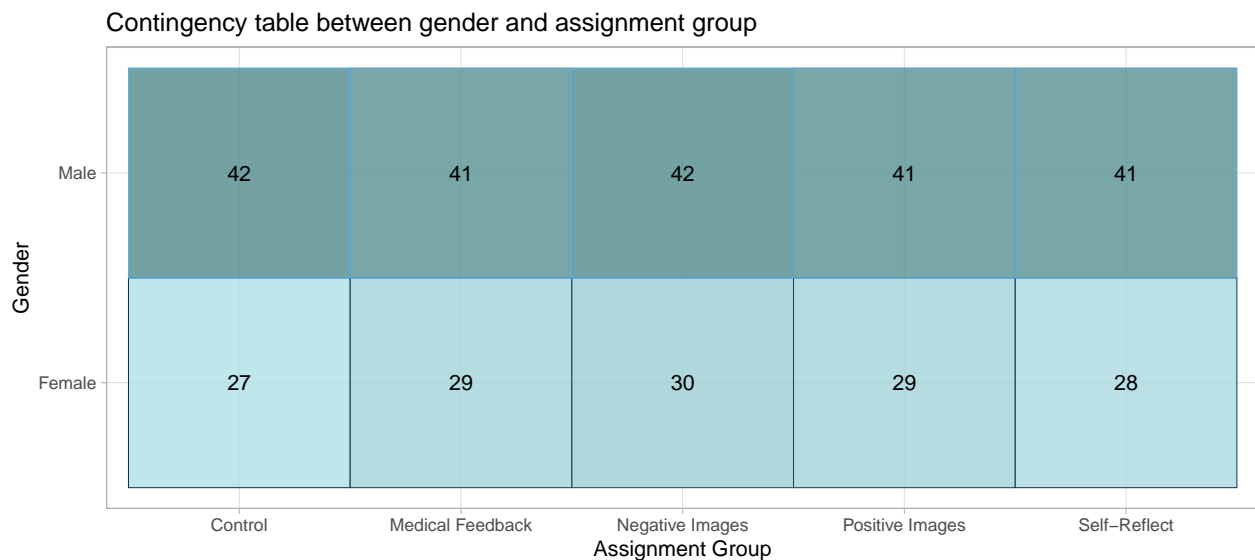
```
# TODO format figures and captions
#check balance between gender
```

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

```
pander(gender_chiq,style='rmarkdown')
```

Quitting from lines 582-598 (EDA.Rmd) Error in pander(gender_chiq, style = "rmarkdown") : object 'gender_chiq' not found Calls: ... withCallingHandlers -> withVisible -> eval -> eval -> pander

```
create_heatmap(var1 = d_respondents$Assignment_Group,var2 = d_respondents$Gender) +
  xlab('Assignment Group') +
  ylab('Gender') +
  labs(title = 'Contingency table between gender and assignment group',
       caption = paste0('Assuming gender distributions are the same among assignment groups, a chi-squared test for independence with 4
                           degrees of freedom yields p=', round(gender_chi$parameter,4), ' \ndegrees of freedom ', 'yields p=',
                           round(gender_chi$p.value,4),
                           ', suggesting that there is no relationship between gender and assignment groups at a
                           significance level of 0.05.'))
  theme(plot.caption = element_text(hjust = 0))
```



Assuming gender distributions are the same among assignment groups, a chi-squared test for independence with 4 degrees of freedom yields $p=0.9981$, suggesting that there is no relationship between gender and assignment groups at a significance level of 0.05.

Age Range

```
# TODO format figures and captions
```

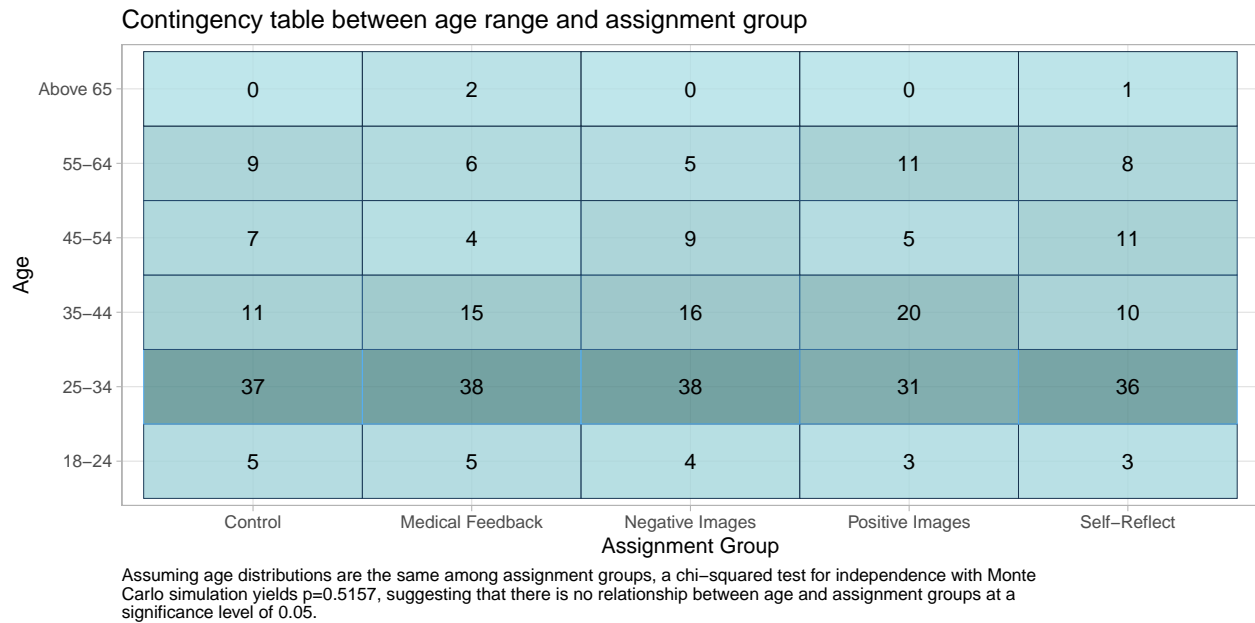
```
#check balance between age-range
```

```
# expected frequency count for each cell of the contingency table should be at least 5. Since this is not the case, we use a simulation-based p-value.
# https://stats.stackexchange.com/questions/81483/warning-in-r-chi-squared-approximation-may-be-incorrect
age_chisq <- chisq.test(d_respondents[, table(Assignment_Group, Age_Range)],simulate.p.value = TRUE)
pander(age_chisq,style='rmarkdown')
```

Table 4: Pearson's Chi-squared test with simulated p-value (based on 2000 replicates): `d_respondents[, table(Assignment_Group, Age_Range)]`

Test statistic	df	P value
19.22	NA	0.5157

```
create_heatmap(var1 = d_respondents$Assignment_Group,var2 = d_respondents$Age_Range) +
  xlab('Assignment Group') +
  ylab('Age') +
  labs(title = 'Contingency table between age range and assignment group',
        caption = paste0('Assuming age distributions are the same among assignment groups, a chi-squared
                           round(age_chisq$p.value,4),
                           ', suggesting that there is no relationship between age and assignment groups a
  theme(plot.caption = element_text(hjust = 0))
```



Education Level

```
# TODO format figures and captions
#check balance between education levels
edu_chisq <- chisq.test(d_respondents[, table(Assignment_Group, Education_Level)],simulate.p.value = T)
pander(edu_chisq,style='rmarkdown')
```

Table 5: Pearson's Chi-squared test with simulated p-value (based on 2000 replicates): `d_respondents[, table(Assignment_Group, Education_Level)]`

Test statistic	df	P value
28.7	NA	0.07146

```
create_heatmap(var1 = d_respondents$Assignment_Group,var2 = d_respondents$Education_Level) +
  xlab('Assignment Group') +
  ylab('Education Level') +
  labs(title = 'Contingency table between education and assignment group',
        caption = paste0('Assuming education distributions are the same among assignment groups, a chi-s
                           round(edu_chisq$p.value,4),
                           ', suggesting that there is no relationship \nbetween education and assignment
  theme(plot.caption = element_text(hjust = 0))
```

Contingency table between education and assignment group

Education Level	Trade school	1	1	3	2	1
	Some high school	0	0	1	0	0
	Master's degree and above	20	14	13	19	11
	High school	1	1	3	0	7
	Bachelor's degree	44	54	50	45	46
	Associate's degree	3	0	2	4	4
		Control	Medical Feedback	Negative Images	Positive Images	Self-Reflect
		Assignment Group				

Assuming education distributions are the same among assignment groups, a chi-squared test for independence with Monte Carlo simulation yields $p=0.0715$, suggesting that there is no relationship between education and assignment groups at a significance level of 0.05.

Country: US, non-US

```
# TODO format figures and captions
# out.width = "80%"
# check balance between US and non-US respondents

us_chisq <- chisq.test(d_respondents[, table(Assignment_Group, US_Dummy)])
pander(us_chisq, style='rmarkdown')
```

Table 6: Pearson's Chi-squared test: `d_respondents[, table(Assignment_Group, US_Dummy)]`

Test statistic	df	P value
6.502	4	0.1647

```
create_heatmap(var1 = d_respondents$Assignment_Group, var2 = d_respondents$US_Dummy) +
  xlab('Assignment Group') +
  ylab('Country') +
  scale_y_discrete(breaks=c("0", "1"),
    labels=c("Non-US", "United States")) +
  labs(title = 'Contingency table between country and assignment group',
    caption = paste0('Assuming country distributions are the same among assignment groups, a chi-squared test with Monte Carlo simulation yields p=',
      round(us_chisq$parameter,4), ' degrees of freedom ', 'yields p=',
      round(us_chisq$p.value,4),
      ', suggesting that there is no relationship between country and assignment \n groups at a significance level of 0.05.'))
  theme(plot.caption = element_text(hjust = 0))
```


Contingency table between country and assignment group

Country	Assignment Group				
	Control	Medical Feedback	Negative Images	Positive Images	Self-Reflect
United States	45	37	45	50	48
Non-US	24	33	27	20	21

Assuming country distributions are the same among assignment groups, a chi-squared test for independence with 4 degrees of freedom yields $p=0.1647$, suggesting that there is no relationship between country and assignment groups at a significance level of 0.05.

```
# ATE of treatment on Total Score
```

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

```
## [1] 0.5143
```

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

```
## [1] 3.743
```

```
# ATE of treatment on TaskPhase2 Score
```

```
d_respondents[ Treatment_Dummy == 1, mean(TaskPhase2_Score)] - d_respondents[ Treatment_Dummy == 0, mean
```

```
## [1] 0.05337
```

```
sd(d_respondents$TaskPhase2_Score)
```

```
## [1] 0.1645
```

```
#trying 2SLS...but dont think it applies here
```

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

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

```
power.t.test( delta = .05, sd=.16, sig.level = 0.05, power=0.8)
```

```
##
```

```
## Two-sample t test power calculation
```

```
##
```

```
## n = 161.7
```

```
## delta = 0.05
```

```
## sd = 0.16
```

```
## 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')),  
  header=FALSE,  
  type='latex')  
  
#add an F test to compare  
pander(anova(mod_task2_a, mod_task2_b, test='F'),style='rmarkdown')
```

Table 8: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
348	9.287	NA	NA	NA	NA
336	8.338	12	0.9498	3.19	0.0002426

```
#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))]
```

Table 7:

	<i>Dependent variable:</i>	
	TaskPhase2_Score	
	(1)	(2)
Treatment_Dummy	0.053** (0.022)	0.051** (0.022)
TaskPhase1_Score		0.240*** (0.047)
as.factor(Gender)Male		-0.010 (0.017)
Constant	0.461*** (0.019)	0.281*** (0.072)
Education Fixed Effects	No	Yes
Age Fixed Effects	No	Yes
Observations	350	350
R ²	0.017	0.117
Adjusted R ²	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

```
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')),
          header=FALSE,
          type='latex')
```

Table 9:

	<i>Dependent variable:</i>	
	TaskPhase2_Score	
	(1)	(2)
as.factor(Assignment_Group)Medical Feedback	0.062** (0.027)	0.055* (0.029)
as.factor(Assignment_Group)Negative Images	0.034 (0.027)	0.039 (0.027)
as.factor(Assignment_Group)Positive Images	0.053* (0.029)	0.050* (0.027)
as.factor(Assignment_Group)Self-Reflect	0.065** (0.028)	0.058** (0.029)
TaskPhase1_Score		0.238*** (0.048)
as.factor(Gender)Male		-0.010 (0.017)
Constant	0.461*** (0.019)	0.282*** (0.073)
Education Fixed Effects	No	Yes
Age Fixed Effects	No	Yes
Observations	350	350
R ²	0.021	0.119
Adjusted R ²	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

```
pander(anova(mod_task2_c, mod_task2_d, test='F'), style='rmarkdown')
```

Table 10: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
345	9.244	NA	NA	NA	NA

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
333	8.323	12	0.921	3.071	0.0003943

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')),
           header=FALSE,
           type='latex')

pander(anova(mod_task3_a, mod_task3_b, test='F'), style='rmarkdown')
```

Table 12: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
348	8.865	NA	NA	NA	NA
336	8.117	12	0.7479	2.58	0.002743

```
# 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')),
           header=FALSE,
           type='latex')

pander(anova(mod_task3_c, mod_task3_d, test='F'), style='rmarkdown')
```

Table 11:

	<i>Dependent variable:</i>	
	TaskPhase3_Score	
	(1)	(2)
Treatment_Dummy	0.004 (0.019)	0.002 (0.019)
TaskPhase1_Score		0.161*** (0.047)
as.factor(Gender)Male		-0.004 (0.017)
Constant	0.538*** (0.017)	0.515*** (0.064)
Education Fixed Effects	No	Yes
Age Fixed Effects	No	Yes
Observations	350	350
R ²	0.0001	0.084
Adjusted R ²	-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

Table 13:

	<i>Dependent variable:</i>	
	TaskPhase3_Score	
	(1)	(2)
as.factor(Assignment_Group)Medical Feedback	0.022 (0.026)	0.011 (0.026)
as.factor(Assignment_Group)Negative Images	-0.015 (0.027)	-0.011 (0.026)
as.factor(Assignment_Group)Positive Images	-0.001 (0.025)	0.004 (0.025)
as.factor(Assignment_Group)Self-Reflect	0.010 (0.026)	0.005 (0.026)
TaskPhase1_Score		0.157*** (0.047)
as.factor(Gender)Male		-0.004 (0.017)
Constant	0.538*** (0.017)	0.518*** (0.064)
Education Fixed Effects	No	Yes
Age Fixed Effects	No	Yes
Observations	350	350
R ²	0.006	0.087
Adjusted R ²	-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)
<i>Note:</i>		
*p<0.1; **p<0.05; ***p<0.01		

Table 14: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
345	8.81	NA	NA	NA	NA
333	8.099	12	0.7113	2.437	0.004751

Wearing Off Effects

```
# TODO add within subjects design
# TODO d_respondents[ , lm(mean(TaskPhase3_Score, TaskPhase2_Score) ~ Assignment_Group + TaskPhase1_Score)]
# TODO d_respondents[ , lm(TaskPhaseB ~ Assignment_Group + TaskPhaseA + as.factor(AmazonTurk_ID))]
mod_task3_e <- d_respondents[ , lm(TaskPhase3_Score ~ TaskPhase2_Score)]
mod_task3_f <- d_respondents[ , lm(TaskPhase3_Score ~ TaskPhase2_Score + Treatment_Dummy)]
mod_task3_g <- d_respondents[ , lm(TaskPhase3_Score ~ TaskPhase2_Score + as.factor(Assignment_Group))]
mod_task3_h <- d_respondents[ , lm(TaskPhase3_Score ~ TaskPhase2_Score +
                                   as.factor(Assignment_Group) +
                                   as.factor(Gender) +
                                   as.factor(Education_Level) +
                                   as.factor(Age_Range))]

stargazer(mod_task3_e,
           mod_task3_f,
           mod_task3_g,
           mod_task3_h,
           se = list(get_robust_se(mod_task3_e),
                     get_robust_se(mod_task3_f),
                     get_robust_se(mod_task3_h)),
                     get_robust_se(mod_task3_g),
           omit = c("Education_Level", "Age_Range"),
           add.lines = list(c('Education Fixed Effects', 'No', 'No', 'No', 'Yes'),
                             c('Age Fixed Effects', 'No', 'No', 'No', 'Yes')),
           header=FALSE,
           type='text')
```

Dependent variable:

TaskPhase3_Score

(1) (2) (3) (4)

TaskPhase2_Score 0.239*** 0.242*** 0.238*** 0.241***
(0.050) (0.051) (0.052) (0.051)

Treatment_Dummy -0.009
(0.019)

as.factor(Assignment_Group)Medical Feedback 0.008 0.001
(0.027) (0.027)

as.factor(Assignment_Group)Negative Images -0.023 -0.023
(0.026) (0.026)

as.factor(Assignment_Group)Positive Images -0.013 -0.007
(0.025) (0.026)

as.factor(Assignment_Group)Self-Reflect -0.005 -0.010

(0.025) (0.027)

as.factor(Gender)Male -0.003
(0.017)

Constant 0.420*** 0.426*** 0.428*** 0.520***
(0.026) (0.028) (0.062) (0.062)

Education Fixed Effects No No No Yes Age Fixed Effects No No No Yes Observations 350 350 350 350 R2
0.061 0.061 0.065 0.113 Adjusted R2 0.058 0.056 0.052 0.070 Residual Std. Error 0.155 (df = 348) 0.155
(df = 347) 0.155 (df = 344) 0.154 (df = 333) F Statistic 22.540*** (df = 1; 348) 11.330*** (df = 2; 347)
4.815*** (df = 5; 344) 2.654*** (df = 16; 333)

=====

Note: $p < 0.1$; $p < 0.05$; $p < 0.01$

=====

(Intercept) TaskPhase2_Score as.factor(Assignment_Group)Medical Feedback
as.factor(Assignment_Group)Negative Images as.factor(Assignment_Group)Positive Images
as.factor(Assignment_Group)Self-Reflect

0.028 0.051 0.026 0.026 0.025 0.024

```
pander(anova(mod_task3_e, mod_task3_f, test='F'), style='markdown')
```

Table 16: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
348	8.326	NA	NA	NA	NA
347	8.322	1	0.004358	0.1817	0.6702

```
pander(anova(mod_task3_g, mod_task3_h, test='F'), style='markdown')
```

Table 17: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
344	8.286	NA	NA	NA	NA
333	7.863	11	0.4227	1.627	0.08955

```
##
mod_task3_i <- d_respondents[, lm(TaskPhase3_Score ~ TaskPhase1_Score + TaskPhase2_Score)]
mod_task3_j <- d_respondents[, lm(TaskPhase3_Score ~ TaskPhase1_Score + TaskPhase2_Score + Treatment_D
mod_task3_k <- d_respondents[, lm(TaskPhase3_Score ~ TaskPhase1_Score + TaskPhase2_Score + as.factor(A
mod_task3_l <- d_respondents[, lm(TaskPhase3_Score ~ TaskPhase1_Score + TaskPhase2_Score +
                                as.factor(Assignment_Group) +
                                as.factor(Gender) +
                                as.factor(Education_Level) +
                                as.factor(Age_Range))]]

stargazer(mod_task3_i,
           mod_task3_j,
           mod_task3_k,
           mod_task3_l,
```

```

se = list(get_robust_se(mod_task3_i),
          get_robust_se(mod_task3_j),
          get_robust_se(mod_task3_k),
          get_robust_se(mod_task3_l)),
header=FALSE,
type='text')

```

=====

Dependent variable:

TaskPhase3_Score

(1) (2) (3) (4)

— TaskPhase1_Score 0.113** 0.113** 0.109** 0.108**
(0.046) (0.046) (0.046) (0.048)

TaskPhase2_Score 0.202*** 0.204*** 0.202*** 0.208***
(0.053) (0.054) (0.053) (0.054)

Treatment_Dummy -0.007
(0.019)

as.factor(Assignment_Group)Medical Feedback 0.007 -0.001
(0.026) (0.026)

as.factor(Assignment_Group)Negative Images -0.019 -0.019
(0.026) (0.026)

as.factor(Assignment_Group)Positive Images -0.012 -0.007
(0.024) (0.025)

as.factor(Assignment_Group)Self-Reflect -0.002 -0.007
(0.024) (0.025)

as.factor(Gender)Male -0.002
(0.016)

as.factor(Education_Level)Bachelor's degree -0.043
(0.046)

as.factor(Education_Level)High school -0.058
(0.054)

as.factor(Education_Level)Master's degree and above -0.065
(0.047)

as.factor(Education_Level)Some high school 0.133***
(0.051)

as.factor(Education_Level)Trade school -0.131**
(0.062)

as.factor(Age_Range)25-34 -0.021
(0.033)

as.factor(Age_Range)35-44 -0.062*
(0.036)

as.factor(Age_Range)45-54 -0.049
(0.043)

as.factor(Age_Range)55-64 -0.056
(0.036)

```
as.factor(Age_Range)Above 65 0.201***
(0.050)

Constant 0.370*** 0.375*** 0.378*** 0.460***
(0.032) (0.034) (0.034) (0.067)
```

```
Observations 350 350 350 350
R2 0.077 0.078 0.081 0.127
Adjusted R2 0.072 0.070 0.064 0.082
Residual Std. Error 0.154 (df = 347) 0.154 (df = 346) 0.154 (df = 343) 0.153 (df = 332)
F Statistic 14.520*** (df = 2; 347) 9.693*** (df = 3; 346) 5.010*** (df = 6; 343) 2.844*** (df = 17; 332)
```

=====

Note: $p < 0.1$; $p < 0.05$; $p < 0.01$

```
pander(anova(mod_task3_i, mod_task3_j, test = 'F'),style='rmarkdown')
```

Table 18: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
347	8.181	NA	NA	NA	NA
346	8.178	1	0.002531	0.1071	0.7437

```
pander(anova(mod_task3_k, mod_task3_l, test = 'F'),style='rmarkdown')
```

Table 19: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
343	8.151	NA	NA	NA	NA
332	7.739	11	0.4128	1.61	0.09435

```
# lm(TaskPhase3_Score ~ TaskPhase2_Score) vs lm(TaskPhase3_Score ~ TaskPhase1_Score + TaskPhase2_Score)
pander(anova(mod_task3_e, mod_task3_i, test = 'F'),style='rmarkdown')
```

Table 20: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
348	8.326	NA	NA	NA	NA
347	8.181	1	0.1455	6.172	0.01345

Playground

```
# compare self-reflect against medical feedback groups?
#make dummies
d_respondents[, Self_Reflect_Dummy := ifelse(Assignment_Group == "Self-Reflect", 1, 0)]
d_respondents[, Med_Feedback_Dummy := ifelse(Assignment_Group == "Medical Feedback", 1, 0)]

mod_test_dummies1 <- d_respondents[, lm(TaskPhase2_Score ~ Treatment_Dummy + Self_Reflect_Dummy)]
mod_test_dummies2 <- d_respondents[, lm(TaskPhase2_Score ~ Treatment_Dummy + Med_Feedback_Dummy)]
```

```

stargazer(mod_test_dummies1,
          mod_test_dummies2,
          se = list(get_robust_se(mod_test_dummies1),
                    get_robust_se(mod_test_dummies2)),
          header=FALSE,
          type = 'latex')

##
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lcc}
## \hline
## \hline \hline
## & \multicolumn{2}{c}{\textit{Dependent variable:}} \hline
## \cline{2-3}
## \hline \hline & \multicolumn{2}{c}{TaskPhase2\_Score} \hline
## \hline \hline & (1) & (2) \hline
## \hline \hline
## Treatment\_Dummy & 0.050$^{**}$ & 0.051$^{**}$ \hline
## & (0.022) & (0.023) \hline
## & & \hline
## Self\_Reflect\_Dummy & 0.016 & \hline
## & (0.023) & \hline
## & & \hline
## Med\_Feedback\_Dummy & & 0.011 \hline
## & & (0.022) \hline
## & & \hline
## Constant & 0.461$^{***}$ & 0.461$^{***}$ \hline
## & (0.019) & (0.019) \hline
## & & \hline
## \hline \hline
## Observations & 350 & 350 \hline
## R$^2$ & 0.018 & 0.017 \hline
## Adjusted R$^2$ & 0.012 & 0.012 \hline
## Residual Std. Error (df = 347) & 0.163 & 0.164 \hline
## F Statistic (df = 2; 347) & 3.192$^{**}$ & 3.079$^{**}$ \hline
## \hline
## \hline \hline
## \textit{Note:} & \multicolumn{2}{r}{\textit{$^*$}$p$<$0.1; \textit{$^{**}$}$p$<$0.05; \textit{$^{***}$}$p$<$0.01} \hline
## \end{tabular}
## \end{table}

# compare positive images against negative images feedback groups?
#make dummies
d_respondents[ , Positive_Images_Dummy := ifelse(Assignment_Group == "Positive Images", 1, 0)]
d_respondents[ , Negative_Images_Dummy := ifelse(Assignment_Group == "Negative Images", 1, 0)]

mod_test_dummies3 <- d_respondents[ , lm(TaskPhase2_Score ~ Treatment_Dummy + Positive_Images_Dummy)]
mod_test_dummies4 <- d_respondents[ , lm(TaskPhase2_Score ~ Treatment_Dummy + Negative_Images_Dummy)]

stargazer(mod_test_dummies3,
          mod_test_dummies4,
          se = list(get_robust_se(mod_test_dummies3),

```

```
get_robust_se(mod_test_dummies4)),
type = 'text')
```

```
##
## =====
##                               Dependent variable:
##                               -----
##                               TaskPhase2_Score
##                               (1)           (2)
## -----
## Treatment_Dummy              0.053**      0.060***
##                               (0.022)      (0.023)
##
## Positive_Images_Dummy         0.0001
##                               (0.024)
##
## Negative_Images_Dummy                -0.027
##                               (0.022)
##
## Constant                      0.461***      0.461***
##                               (0.019)      (0.019)
## -----
## Observations                  350           350
## R2                           0.017           0.021
## Adjusted R2                   0.011           0.015
## Residual Std. Error (df = 347) 0.164           0.163
## F Statistic (df = 2; 347)       2.947*          3.670**
## =====
## Note:                        *p<0.1; **p<0.05; ***p<0.01
```

Playground 2

```
### 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 t value
## (Intercept) 0.28108 0.07213 3.90
## Treatment_Dummy 0.05071 0.02217 2.29
## TaskPhase1_Score 0.24027 0.04682 5.13
## as.factor(Education_Level)Bachelor's degree -0.00683 0.04856 -0.14
## as.factor(Education_Level)High school 0.04068 0.05619 0.72
## as.factor(Education_Level)Master's degree and above -0.01698 0.05128 -0.33
## as.factor(Education_Level)Some high school -0.12065 0.05108 -2.36
## as.factor(Education_Level)Trade school 0.02867 0.06926 0.41
## as.factor(Gender)Male -0.00995 0.01735 -0.57
## as.factor(Age_Range)25-34 0.04469 0.03768 1.19
## as.factor(Age_Range)35-44 0.04198 0.03952 1.06
## as.factor(Age_Range)45-54 0.06975 0.04178 1.67
## as.factor(Age_Range)55-64 0.08035 0.04252 1.89
## as.factor(Age_Range)Above 65 0.12575 0.05172 2.43
##
## Pr(>|t|)
## (Intercept) 0.00012 ***
## Treatment_Dummy 0.02281 *
## TaskPhase1_Score 4.9e-07 ***
## as.factor(Education_Level)Bachelor's degree 0.88827
## as.factor(Education_Level)High school 0.46957
## as.factor(Education_Level)Master's degree and above 0.74080
## as.factor(Education_Level)Some high school 0.01874 *
## as.factor(Education_Level)Trade school 0.67922
## as.factor(Gender)Male 0.56658
## as.factor(Age_Range)25-34 0.23642
## as.factor(Age_Range)35-44 0.28890
## as.factor(Age_Range)45-54 0.09593 .
## as.factor(Age_Range)55-64 0.05968 .
## as.factor(Age_Range)Above 65 0.01556 *
## ---
## 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.
## 0.200 0.500 0.600 0.606 0.700 1.000
```

```
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='latex')
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
## % Date and time: Sun, Dec 06, 2020 - 13:51:13
## \begin{table}[!htbp] \centering
## \caption{}
## \label{}
##
```

```

## \begin{tabular}{@{\extracolsep{5pt}}lccc}
## \[-1.8ex]\hline
## \hline \[-1.8ex]
## & \multicolumn{3}{c}{\textit{Dependent variable:}} \\
## \cline{2-4}
## \[-1.8ex] & \multicolumn{3}{c}{TaskPhase2\_Score} \\
## \[-1.8ex] & (1) & (2) & (3) \\
## \hline \[-1.8ex]
## TaskPhase1\_Score & 0.249$^{***}$ & 0.153 & 0.240$^{***}$ \\
## & (0.044) & (0.095) & (0.047) \\
## & & & \\
## as.factor(Education\_Level)Bachelor's degree & & & $-0.007 \\
## & & & (0.049) \\
## & & & \\
## as.factor(Education\_Level)High school & & & 0.041 \\
## & & & (0.056) \\
## & & & \\
## as.factor(Education\_Level)Master's degree and above & & & $-0.017 \\
## & & & (0.051) \\
## & & & \\
## as.factor(Education\_Level)Some high school & & & $-0.121$^{**}$ \\
## & & & (0.051) \\
## & & & \\
## as.factor(Education\_Level)Trade school & & & 0.029 \\
## & & & (0.069) \\
## & & & \\
## as.factor(Gender)Male & & & $-0.010 \\
## & & & (0.017) \\
## & & & \\
## as.factor(Age\_Range)25-34 & & & 0.045 \\
## & & & (0.038) \\
## & & & \\
## as.factor(Age\_Range)35-44 & & & 0.042 \\
## & & & (0.040) \\
## & & & \\
## as.factor(Age\_Range)45-54 & & & 0.070$^{*}$ \\
## & & & (0.042) \\
## & & & \\
## as.factor(Age\_Range)55-64 & & & 0.080$^{*}$ \\
## & & & (0.043) \\
## & & & \\
## as.factor(Age\_Range)Above 65 & & & 0.126$^{**}$ \\
## & & & (0.052) \\
## & & & \\
## Treatment\_Dummy & 0.054$^{**}$ & $-0.019 & 0.051$^{**}$ \\
## & (0.021) & (0.065) & (0.022) \\
## & & & \\
## TaskPhase1\_Score:Treatment\_Dummy & & 0.120 & \\
## & & (0.107) & \\
## & & & \\
## Constant & 0.310$^{***}$ & 0.368$^{***}$ & 0.281$^{***}$ \\
## & (0.032) & (0.057) & (0.072) \\
## & & & \\
## \hline \[-1.8ex]

```

```
## Observations & 350 & 350 & 350 \\
## R2 & 0.098 & 0.101 & 0.117 \\
## Adjusted R2 & 0.092 & 0.093 & 0.083 \\
## Residual Std. Error & 0.157 (df = 347) & 0.157 (df = 346) & 0.158 (df = 336) \\
## F Statistic & 18.780*** (df = 2; 347) & 12.920*** (df = 3; 346) & 3.433*** (df = 13; 336) \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:} & \multicolumn{3}{r}{*p<$0.1; **p<$0.05; ***p<$0.01} \\
## \end{tabular}
## \end{table}
```

```
mod_test4 <- d_test[ , lm(TaskPhase3_Score ~ TaskPhase2_Score)]
coeftest(mod_test4)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.4205    0.0267   15.77 <2e-16 ***
## TaskPhase2_Score 0.2389    0.0503    4.75  3e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# use Robust SE
```

```
mod_test2 <- d_respondents[, lm(TaskPhase2_Score ~ Treatment_Dummy + as.factor(Education_Level) + (Treatment_Dummy:as.factor(Education_Level)))]
mod_test2$vcovHC_ <- vcovHC(mod_test2)
coeftest(mod_test2, vcov = mod_test2$vcovHC_)
```

```
##
## t test of coefficients:
##
##              Estimate
## (Intercept)      0.53333
## Treatment_Dummy    0.00667
## as.factor(Education_Level)Bachelor's degree -0.07424
## as.factor(Education_Level)High school -0.03333
## as.factor(Education_Level)Master's degree and above -0.07333
## as.factor(Education_Level)Some high school -0.14000
## as.factor(Education_Level)Trade school -0.23333
## Treatment_Dummy:as.factor(Education_Level)Bachelor's degree 0.04142
## Treatment_Dummy:as.factor(Education_Level)High school 0.07515
## Treatment_Dummy:as.factor(Education_Level)Master's degree and above 0.04386
## Treatment_Dummy:as.factor(Education_Level)Trade school 0.30762
##              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

```