title: "Moving the Needle on Public Opinion: An Experiment on the Persuasive Effects of Moral Frames"
subtitle: "W241 Experiments and Causality"
author: "Kevin Hartman, Hanna Rocks, Tim Spittle, and Jay Venkata"
date: "December 10, 2019"
tags: [Moral Foundations, Universal Basic Income, Survey, Experiment]
abstract: Through this experiment we tested the treatment effect of various presentations of the moral foundations ("the frame") on a person's feelings towards a particular topic.
output:
pdf_document:
fig_caption: yes
number_sections: yes
toc: no
latex_engine: xelatex
md_document:
variant: markdown
html_document:
df_print: paged
toc: yes

bibliography: bibliography.bib

```{r setup, include = FALSE}
rm(list=ls())

packages = c('openxlsx'
, 'tidyverse', 'data.table'
, 'visNetwork'
, 'lmtest', 'sandwich', 'car', 'survey'
, 'gridExtra', 'stargazer', 'cowplot', 'corrplot'
, 'knitr', 'webshot'
, 'png', 'grid')
packages_needed = packages[!packages %in% installed.packages()]
if(length(packages_needed) > 0) install.packages(packages_needed)
lapply(packages, library, character.only = TRUE)

stargazer_type = "latex" #"latex" # change to latex when ready to knit
```

{r import, include = FALSE}

Survey Results

Panel 1

results_raw_panel1 = read.csv("./data/study/MF Framing Pilot - Full Recruitment - Panel 1.csv", stringsAsFactors = FALSE) %>%
filter(!grepl("Start|Import", StartDate)) %>%
mutate(panel = 1)

Panel 2

results_raw_panel2 = read.csv("./data/study/MF Framing Pilot - Full Recruitment - Panel 2.csv", stringsAsFactors = FALSE) %>%
filter(!grepl("Start|Import", StartDate)) %>%
mutate(panel = 2)

Panel 2 - 10 Control Females

results_raw_panel2_10fem = read.csv("./data/study/MF Framing Pilot - Full Recruitment - Panel 2 - 10 Female Control.csv", stringsAsFactors = FALSE) %>%
filter(!grepl("Start|Import", StartDate)) %>%
mutate(panel = 2)

Participant Details

Panel 1

participant_detail_panel1 = read.csv("./data/study/Prolific Participants - Panel 1.csv", stringsAsFactors = FALSE)

Panel 2

participant_detail_panel2 = read.csv("./data/study/Prolific Participants - Panel 2.csv", stringsAsFactors = FALSE)

Panel 2 - 10 Control Females

participant_detail_panel2_10fem = read.csv("./data/study/Prolific Participants - Panel 2 - 10 Female Control.csv", stringsAsFactors = FALSE)

{r deduplicate, include = FALSE}

Stack panel data (500 obs x 63 vars)

```
results_stacked = bind_rows(results_raw_panel1
, results_raw_panel2
, results_raw_panel2_10fem
) %>%
# Duplicate responses are a product of our multiple data extracts, remove as-is 500 obs x 63 vars)
distinct(ResponseId, .keep_all= TRUE)
```

Identify if the same person filled out the survey >1x

```
results_ids_dedup = results_stacked %>%
select(PROLIFIC_PID, ResponseId, StartDate) %>%
group_by(PROLIFIC_PID) %>%
summarize(count = n()
, min_date = min(StartDate))
```

Keep only their first submission (500 obs x 64 vars [count will tell us who filled out 2x])

```
results_dedup = results_stacked %>%
merge(results_ids_dedup
, by.x = c("PROLIFIC_PID", "StartDate")
, by.y = c("PROLIFIC_PID", "min_date")
, all.x = TRUE) %>%
filter(!is.na(count))
```

Stack participant details (508 obs x 21 vars)

```
participaipant_detail_stacked = bind_rows(participant_detail_panel1
, participant_detail_panel2
, participant_detail_panel2_10fem) %>%
# Duplicate responses are a product of our multiple data extracts, remove as-is (508 obs x 21 vars)
distinct(session_id, .keep_all= TRUE) %>%
# Remove vars that will be in results as well (518 obs x 19 vars)
select(-status, -age)
```

Identify if the same person filled out the survey >1x

```
participaipant_ids_dedup = participaipant_detail_stacked %>%
```

```
select(participant_id, session_id, started_datetime) %>%
group_by(participant_id) %>%
summarize(count = n()
, min_date = min(started_datetime))
```

Keep only their first submission (508 obs x 18 vars [count will tell us who filled out 2x])

```
participaipant_detail_dedup = participaipant_detail_stacked %>%
merge(participaipant_ids_dedup
, by.x = c("participant_id", "started_datetime")
, by.y = c("participant_id", "min_date")
, all.x = TRUE) %>%
filter(!is.na(count)) %>% select(-count, -session_id)
```

Merge for final dataset (500 obs x 81 vars)

```
results_full_dedup = merge(results_dedup
, participaipant_detail_dedup
, by.x = "PROLIFIC_PID"
, by.y = "participant_id"
, all.x = TRUE)
```

{r data_cleaning, include = FALSE}

Adjust all variable names to remove '-' and '.' + lowercase

```
names(results_full_dedup) = tolower(gsub(x = names(results_full_dedup), pattern = "\-|\.",
replacement = "_"))
```

Discrete variables as factors (manual ordering for plotting)

```
arm_levels = c("Control", "Purity_Base", "Purity_Extension", "Fairness_Base",
"Fairness_Extension")
ideology_levels = c("Very Liberal", "Lean Liberal", "Liberal", "Moderate", "Conservative", "Lean
Conservative", "Very Conservative")
response_levels = c("None at all", "A little", "A moderate amount", "A lot", "A great deal")
ubi_group_levels = c("Promoter", "Passive", "Detractor")
ubi_familiarity_levels = c("Extremely familiar", "Very familiar", "Moderately familiar", "Slightly
familiar", "Not familiar at all")
recruit_wave_levels = c("Wave1", "Wave2", "Wave3", "Wave4", "Wave5")

results_full = results_full_dedup %>%
# Define arms and nodes
mutate(arm = case_when(grepl('a', fc_b_1, ignore.case = TRUE) ~ "Purity_Base"
, grepl('a', fc_c_1, ignore.case = TRUE) ~ "Purity_Extension"
, grepl('a', fc_d_1, ignore.case = TRUE) ~ "Fairness_Base"
, grepl('a', fc_e_1, ignore.case = TRUE) ~ "Fairness_Extension"
, TRUE ~ "Control") %>% factor(levels = arm_levels)
, node = paste0(arm, "panel", panel)
, arm_level = case_when(grepl('Base', arm) ~ 'Base'
, grepl('Extension', arm) ~ 'Extension'
, TRUE ~ 'Control') %>% factor(levels = c("Control", "Base", "Extension"))
, arm_story = case_when(grepl('Purity', arm) ~ 'Purity'
, grepl('Fairness', arm) ~ 'Fairness'
, TRUE ~ 'Control') %>% factor(levels = c("Control", "Purity", "Fairness"))
# Combine reaction vars from different arms
, purity_q1_self = case_when(grepl('a', fc_b_1, ignore.case = TRUE) ~ fc_b_1
, grepl('a', fc_c_1, ignore.case = TRUE) ~ fc_c_1
, TRUE ~ NA_character_)
, purity_q2_repulsed = case_when(grepl('a', fc_b_2, ignore.case = TRUE) ~ fc_b_2
, grepl('a', fc_c_2, ignore.case = TRUE) ~ fc_c_2
, TRUE ~ NA_character_)
, purity_q3_injustice = case_when(grepl('a', fc_b_3, ignore.case = TRUE) ~ fc_b_3
, grepl('a', fc_c_3, ignore.case = TRUE) ~ fc_c_3
, TRUE ~ NA_character_)
```

```
, purity_q4_relieved = case_when(grepl('a', fc_c_4, ignore.case = TRUE) ~ fc_c_4
, TRUE ~ NA_character_)
, fairness_q1_self = case_when(grepl('a', fc_d_1, ignore.case = TRUE) ~ fc_d_1
, grepl('a', fc_e_1, ignore.case = TRUE) ~ fc_e_1
, TRUE ~ NA_character_)
, fairness_q2_pain = case_when(grepl('a', fc_d_2, ignore.case = TRUE) ~ fc_d_2
, grepl('a', fc_e_2, ignore.case = TRUE) ~ fc_e_2
, TRUE ~ NA_character_)
, fairness_q3_injustice = case_when(grepl('a', fc_d_3, ignore.case = TRUE) ~ fc_d_3
, grepl('a', fc_e_3, ignore.case = TRUE) ~ fc_e_3
, TRUE ~ NA_character_)
, fairness_q4_relieved = case_when(grepl('a', fc_e_4, ignore.case = TRUE) ~ fc_e_4
, TRUE ~ NA_character_)
# Bin reaction vars
, purity_q2_repulsed_bin = case_when(is.na(purity_q2_repulsed) ~ NA_real_
, purity_q2_repulsed %in% c("None at all", "A little") ~ 0
, purity_q2_repulsed %in% c("A moderate amount", "A great deal", "A lot") ~ 1
, TRUE ~ NA_real_) %>% factor()
, purity_q4_relieved_bin = case_when(is.na(purity_q4_relieved) ~ NA_real_
, purity_q4_relieved %in% c("None at all", "A little") ~ 0
, purity_q4_relieved %in% c("A moderate amount", "A great deal", "A lot") ~ 1
, TRUE ~ NA_real_) %>% factor()
, fairness_q2_pain_bin = case_when(is.na(fairness_q2_pain) ~ NA_real_
, fairness_q2_pain %in% c("None at all", "A little") ~ 0
, fairness_q2_pain %in% c("A moderate amount", "A great deal", "A lot") ~ 1
, TRUE ~ NA_real_) %>% factor()
, fairness_q4_relieved_bin = case_when(is.na(fairness_q4_relieved) ~ NA_real_
, fairness_q4_relieved %in% c("None at all", "A little") ~ 0
, fairness_q4_relieved %in% c("A moderate amount", "A great deal", "A lot") ~ 1
, TRUE ~ NA_real_) %>% factor()
, open_text_reaction = q3_fc2
# Factor variables
, ideology = factor(polispect, levels = ideology_levels)
, ideology_bin = case_when(is.na(ideology) ~ "missing"
, ideology == "Very Liberal" ~ "Liberal"
, ideology == "Lean Liberal" ~ "Liberal"
, ideology == "Liberal" ~ "Liberal"
, ideology == "Very Conservative" ~ "Conservative"
, ideology == "Lean Conservative" ~ "Conservative"
```

```
, ideology == "Conservative" ~ "Conservative"
, TRUE ~ "Moderate")
# UBI/Outcome
, ubi_group = factor(ubi_2_nps_group, levels = ubi_group_levels)
, ubi_familiarity = factor(ubi_f, levels = ubi_familiarity_levels)
, ubi_familiarity_bin = case_when(is.na(ubi_f) ~ NA_real_
, ubi_f == "Not familiar at all" ~ 0
, TRUE ~ 1) %>% factor()
, ubi_number = as.numeric(ubi_2)
# Recruitment Day blocks
, recruitment_wave = case_when(is.na(recruitday) ~ "missing"
, recruitday == "T1" ~ "Wave1"
, recruitday == "F" ~ "Wave2"
, recruitday == "SU" ~ "Wave3"
, recruitday == "M" ~ "Wave4"
, recruitday == "T2" ~ "Wave5"
, TRUE ~ "unknown") %>% factor(levels = recruit_wave_levels)
)
```

Clean = limit to the variables we need

```
results_clean = results_full %>%
select(prolific_pid, panel, arm, node, arm_level, arm_story
, ideology, ideology_bin, age, gender, urban, employment_status, student_status
, purity_q1_self, purity_q2_repulsed, purity_q3_injustice, purity_q4_relieved
, fairness_q1_self, fairness_q2_pain, fairness_q3_injustice, fairness_q4_relieved
, purity_q2_repulsed_bin, purity_q4_relieved_bin, fairness_q2_pain_bin, fairness_q4_relieved_bin
, open_text_reaction
, ubi_number, ubi_group, ubi_familiarity, ubi_familiarity_bin, recruitday, recruitment_wave)
```

Arm-specific datasets

```
results_armlibfair = results_clean %>% filter(ideology_bin == 'Liberal' & grepl('Fairness|Control', arm))
results_armlibpure = results_clean %>% filter(ideology_bin == 'Liberal' & grepl('Purity|Control', arm))
```

results_armconfair = results_clean %>% filter(ideology_bin == 'Conservative' &
grepl('Fairness|Control', arm))
results_armconpure = results_clean %>% filter(ideology_bin == 'Conservative' &
grepl('Purity|Control', arm))

Remove moderates for EDA

results_clean_lim = results_clean %>% filter(ideology_bin != "Moderate")

Controls only for recruitment day test

results_clean_lim_ctrl = results_clean %>% filter(ideology_bin != "Moderate" & grepl('Control',
arm))

{r lm_function, include = FALSE}
custom_lm_calcs = function(lm_in, clusters_in){
# Robust
vcov_robust = vcovHC(lm_in)
se_robust = sqrt(diag(vcov_robust))
# Cluster
if(length(clusters_in) > 1){
vcov_cluster = cluster.vcov(lm_in, clusters_in)
se_cluster = sqrt(diag(vcov_cluster))
} else {
vcov_cluster = NA
se_cluster = NA
}
# Output
lm_out = list(lm = lm_in
, vcov_robust = vcov_robust
, se_robust = se_robust
, vcov_cluster = vcov_cluster
, se_cluster = se_cluster

)
return(lm_out)
}

```
# Introduction
We make hundreds of decisions each day. We may spend minutes, even hours, considering

In the early 2000s, psychologists Jonathan Haidt, Craig Joseph and Jesse Graham propos

Moral foundations theory has often been applied to studies of political science, diffe

Perhaps in a perfect world, every voter would decide on issues and candidates based on

Perhaps in a perfect world, every voter would decide on issues and candidates based on

# Background & Motivation

Online marketing campaigns and social media has increased the specificity with which p

Day et al executed two experiments to test the effects of moral foundation-based "fram

Both studies supported the hypothesis that an individual's political attitude is bolst

Another study that played a key role in defining this experiment was a paper by Lene A

To build on the conclusions of Day et al's work, we designed a study to measure the ef

# Experimental Design

## Hypothesis

Can appealing to one or more of the moral foundations modify how an individual feels a

## Treatment

To test this hypothesis, we had to define the frames for our experiment. A "frame" is

We proceeded to develop frames in the form of written vignettes with one to two relate

As is described in greater detail in the Procedure section of this report, we scoped o

As is evident when reading Entman's description of a frame, it is a somewhat "fuzzy" c

## Distribution

_[Not Updated as of Dec 13 @ 3 PM]_

Given the numerous arms of our proposed study (described in detail in Experiment Desig
```

```
Prolific also facilitated blocking to obtain an equal distribution of participants who

Of the 505 subjects who participated in the study, all but three from the conservative

Our study specifically targeted subjects who identified as either politically liberal

The subject pool comprised of XX (x%) .. Figure 1 shows further detail on the politica

Collection of participants took place over several days. We had limited funds availabl

Overall, we had X waves. Table 1 provides additional information on each wave. Overall

We have a high level of confidence in the randomization of our effective participant p
```

{r exploratory_demographics, echo = FALSE, fig.height = 8, fig.width = 8, fig.align="center", fig.cap="Demographics"}

Ideology

plot_ideology = ggplot() +
geom_bar(data = results_clean_lim %>% group_by(ideology) %>%
summarise(Count = n()) %>% rename("Ideology" = ideology)
, aes(x = Ideology, y = Count, fill = Ideology), stat = "identity") +
scale_fill_brewer(type = "div", palette = 5, direction = -1, aesthetics = "fill") +
theme_bw() +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
theme(legend.position = "left", axis.text.x=element_blank(), legend.text=element_text(size = 8))

Backbone ggplot object

grpstackbar_plot = ggplot() +
facet_grid(~ ideology_bin) +
scale_fill_brewer(type = "div", palette = 5, direction = -1, aesthetics = "fill") +
theme_bw() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

Age

```
plot_age = grpstackbar_plot +
geom_bar(data = results_clean_lim %>% group_by(age, ideology, ideology_bin) %>%
summarise(Count = n()) %>% rename("Ideology" = ideology, "Age" = age)
, aes(x = Age, y = Count, fill = Ideology), stat = "identity", show.legend = FALSE)
```

Gender

```
plot_gender = grpstackbar_plot +
geom_bar(data = results_clean_lim %>% group_by(gender, ideology, ideology_bin) %>%
summarise(Count = n()) %>% rename("Ideology" = ideology, "Gender" = gender)
, aes(x = Gender, y = Count, fill = Ideology), stat = "identity", show.legend = FALSE)
```

Urban

```
plot_urban = grpstackbar_plot +
geom_bar(data = results_clean_lim %>% group_by(urban, ideology, ideology_bin) %>%
summarise(Count = n()) %>% rename("Ideology" = ideology, "Urban" = urban)
, aes(x = Urban, y = Count, fill = Ideology), stat = "identity", show.legend = FALSE)
```

Histogram of familiarity

```
plot_familiarity = grpstackbar_plot +
geom_bar(data = results_clean_lim %>% filter(!is.na(ubi_familiarity)) %>%
group_by(ubi_familiarity, ideology, ideology_bin) %>%
summarise(Count = n()) %>%
rename("Ideology" = ideology, "UBI Familiarity" = ubi_familiarity)
, aes(x = `UBI Familiarity`, y = Count, fill = Ideology), stat="identity", show.legend = FALSE)
```

PLOT

```
grid.arrange(plot_ideology, plot_gender
, plot_age, plot_familiarity
, nrow = 2)
```

```
{r exploratory_reactions, echo = FALSE, fig.height = 8, fig.width = 8, fig.align="center",
fig.cap="Reactions"}
results_response = results_clean_lim %>%
mutate("See myself" = case_when(arm_story == "Purity" ~ purity_q1_self, TRUE ~
fairness_q1_self)
, "Felt repulsed/pain" = case_when(arm_story == "Purity" ~ purity_q2_repulsed, TRUE ~
fairness_q2_pain)
, "Felt relieved" = case_when(arm_story == "Purity" ~ purity_q4_relieved, TRUE ~
fairness_q4_relieved)) %>%
select(ideology_bin, arm_level, arm_story, `See myself`, `Felt repulsed/pain`, `Felt relieved`)
%>%
gather(prompt, value, -ideology_bin, -arm_level, -arm_story) %>%
filter(!is.na(value)) %>%
group_by(ideology_bin, arm_level, arm_story, prompt, value) %>%
summarise(count = n()) %>%
mutate(Response = factor(value, levels = response_levels)) %>%
group_by(ideology_bin, arm_level, arm_story, prompt) %>%
mutate(count_total_cohort = sum(count)
, Share = count/count_total_cohort
, Prompt = factor(prompt, levels = c("See myself", "Felt repulsed/pain", "Felt relieved")))

ggplot(data = results_response
, aes(x = Prompt, y = Response, fill = Share)) +
geom_tile() +
facet_grid(rows = vars(arm_story, arm_level)
, cols = vars(ideology_bin)) +
scale_fill_distiller(direction = 1, palette = 2) +
theme_bw() +
theme(axis.text.x = element_text(angle = 45, hjust = 1)
, legend.position = "none")
```

```
\newpage
```

```
## Procedure

We developed two treatment options for both moral foundations included in our experime

All subjects assigned to treatment would read a "base" story designed to trigger eithe

We implemented a timer on the treatment pages of the survey requiring participants to

Figure X offers a detailed description of the four treatment arms and control arm for

After navigating through treatment or control, the subjects were asked to share their

All participants answered the following demographic questions before concluding the su

Participants in the pilot study and select waves were also asked about their reactions

See _\autoref{fig:study_flowchart}_ for detailed flowchart of study design.

## Modifications

Our first wave of participants comprised of 100 conservatives and 100 liberals split e

We noted the greatest movement amongst the conservatives being treated with the purity

In a world with limitless funds, we would have been able to collect enough participant

# Analysis of Results

## Data

Given our "waves" approach to data collection, we wanted to ensure the wave/day blocki

## Models
```

{r model_all_arms, echo = FALSE, results='asis'}
model_libfair = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level, data = results_armlibfair), clusters_in = NA)
model_libpure = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level, data = results_armlibpure), clusters_in = NA)
model_confair = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level, data = results_armconfair), clusters_in = NA)
model_conpure = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level, data = results_armconpure), clusters_in = NA)

stargazer(model_libfair$lm, model_libpure$lm

, model_confair$lm, model_conpure$lm
, se = list(model_libfair$se_robust, model_libpure$se_robust
, model_confair$se_robust, model_conpure$se_robust)
, report = ('vcsp')
, header = F
, title = "By Arm"
, dep.var.caption = "Four Study Arms"
, dep.var.labels = "UBI Ranking"
, column.labels = c("Lib + Fair", "Lib + Pure"
, "Con + Fair", "Con + Pure")
, covariate.labels = c("Base Only Treatment", "Base + Extension Treatment")
, notes = c("HC Robust Standard Errors"
, "Lib = Liberal | Con = Conservative"
, "Pure = Purity Frame | Fair = Fairness Frame")
, font.size = "small"
, column.sep.width = "1pt"
, label = "tab:allarmsmodel"
, type = stargazer_type
)

```
_\autoref{tab:allarmsmodel})_ presents the results of all treatments collected. Note t

The results demonstrate that the extension of the Purity foundation story, when shown

The Base (negative) implementation of purity showed only minimal upward influence but

Liberal baseline opinions of UBI were so high to begin with that our hypothesized infl

\newpage
```

{r model_conpure_interactions, echo = FALSE, results='asis'}

Gender

model_conpure_gender = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level + gender
, data = results_armconpure)
, clusters_in = NA)

Familiarity

```
model_conpure_familiarity = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level +
ubi_familiarity_bin
, data = results_armconpure)
, clusters_in = NA)
```

Reaction (need to re-factor reactions first)

```
results_armconpure_reaction = results_armconpure %>%
mutate(purity_q2_repulsed_bin_f = case_when(is.na(purity_q2_repulsed_bin) ~ 0
, TRUE ~ as.numeric(as.character(purity_q2_repulsed_bin))) %>% factor()
, purity_q4_relieved_bin_f = case_when(is.na(purity_q4_relieved_bin) ~ 0
, TRUE ~ as.numeric(as.character(purity_q4_relieved_bin))) %>% factor()
)
model_conpure_reaction = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level +
purity_q2_repulsed_bin_f * purity_q4_relieved_bin_f
, data = results_armconpure_reaction)
, clusters_in = NA)

stargazer(model_conpure$lm
, model_conpure_gender$lm
, model_conpure_familiarity$lm
, model_conpure_reaction$lm
, se = list(model_conpure$se_robust
, model_conpure_gender$se_robust
, model_conpure_familiarity$se_robust
, model_conpure_reaction$se_robust
)
, report = ('vcsp')
, header = F
, title = "Conservative + Purity Treatment Arm Interaction Specifications"
, dep.var.caption = "Con + Pure Arm Only"
, dep.var.labels = "UBI Ranking"
, column.labels = c("No Covariates", "Gender", "UBI Familiarity", "Reaction")
, covariate.labels = c("Base Only Treatment", "Base + Extension Treatment"
, "Male", "Familiar w/ UBI"
```

, "Repulsed", "Relieved", "Repulsed then Relieved")
, notes = c("HC Robust Standard Errors"
, "Lib = Liberal | Con = Conservative"
, "Pure = Purity Frame | Fair = Fairness Frame")
, font.size = "small"
, column.sep.width = "1pt"
, label = "tab:covariatesmodel"
, type = stargazer_type
)

```
Given our focus on the Conservative Purity arms, we also tested covariate specificatio

Gender has a significant difference at baseline (`ubi_number ~ arm_level + gender`): m

Being familiar with UBI was associated with a slightly lower, though statistically ins

Given that neither of our hypothesized baseline differences from gender of UBI familia

\newpage

# Conclusion

## Discussion

Our experiment demonstrates that it may be possible to affect attitudes toward politic

## Limitations

Isolating effect attributable to a specific moral foundation is not feasible
Any stimuli can trigger multiple moral foundations
In effect, this experiment only tests the effectiveness of our stimuli in effecting th
In practical terms, while this finding may seem to be immediately applicable in analog
Many stimuli and targets would need to be tested while controlling for nuances between

\pagebreak

# Appendix

## Declaration of Conflicting Interests

To the best of their knowledge, the authors have no potential conflicts of interest wi

## Funding

The authors received $500 in financial support from the University of California, Berk

In addition, the authors put in $25 out of their own personal income to increase the s
```

```
## Study Flowchart
```

{r flowchart, echo = FALSE, fig.height = 8, fig.width = 8, fig.align="center",
fig.cap="\label{fig:study_flowchart} Study Flowchart"}
flowchart_img = readPNG("images/flowchart.png")
grid.raster(flowchart_img)

```
## Data Dictionary

| Variable Name | Variable | Values |
| :---- | :-------- | :-------- |
| `prolific_pid` | User ID | 24-digit alphanumeric |
| `arm_story` | Treatment Story | Control, Purity, or Fairness |
| `arm_level` | Treatment Level | Control, Base, or Extension |
| `ideology_bin` | Ideology| Conservative or Liberal |
| `ubi_number` | UBI Ranking | 0 (least) - 10 (most) support for UBI |
| `gender` | Gender | Male or Female |
| `ubi_familiarity_bin` | UBI Familiarity | 0 (None/A little) or 1 (Any higher familia
| `purity_q2_repulsed_bin` | Repulsed | 0 (None/A little) or 1 (Any higher reaction) |
| `purity_q4_relieved_bin` | Relieved | 0 (None/A little) or 1 (Any higher reaction) |

\pagebreak

## Treatment Vignettes

_[[TBD]]_
```

{r vignette_purity_images_base, echo = FALSE, fig.show='hold', out.width = "50%", fig.align = "default"}

EXAMPLE of how to include images - rest of the images to be added with text

knitr::include_graphics("images/vignette_purity_base1.png")
knitr::include_graphics("images/vignette_purity_base2.png")

```
## Covariate Balance Check
```

{r model_arm_cov_balaance, echo = FALSE, results='asis'}
arm_pur_base = custom_lm_calcs(lm_in = lm(arm=="Purity_Base" ~ age + gender + urban, data = results_clean_lim %>%
filter(recruitment_wave %in% c("Wave1"), ideology_bin %in%("Conservative")))#, "Wave2")))
, clusters_in = NA)
arm_pur_ext = custom_lm_calcs(lm_in = lm(arm=="Purity_Extension" ~ age + gender + urban, data = results_clean_lim %>%
filter(recruitment_wave %in% c("Wave1"), ideology_bin %in%("Conservative")))#, "Wave2")))
, clusters_in = NA)
arm_fair_base = custom_lm_calcs(lm_in = lm(arm=="Fairness_Base" ~ age + gender + urban, data = results_clean_lim %>%
filter(recruitment_wave %in% c("Wave1"), ideology_bin %in%("Conservative")))#, "Wave2")))
, clusters_in = NA)
arm_fair_ext = custom_lm_calcs(lm_in = lm(arm=="Fairness_Extension" ~ age + gender + urban, data = results_clean_lim %>%
filter(recruitment_wave %in% c("Wave1"), ideology_bin %in%("Conservative")))#, "Wave2")))
, clusters_in = NA)
arm_control = custom_lm_calcs(lm_in = lm(arm=="Control" ~ age + gender + urban, data = results_clean_lim %>%
filter(recruitment_wave %in% c("Wave1"), ideology_bin %in%("Conservative")))#, "Wave2")))
, clusters_in = NA)

stargazer(arm_pur_base$lm
, arm_pur_ext$lm
, arm_fair_base$lm
, arm_fair_ext$lm
, arm_control$lm
, se = list(arm_pur_base$se_robust
, arm_pur_ext$se_robust
, arm_fair_base$se_robust
, arm_fair_ext$se_robust
, arm_control$se_robust)
, report = ('vcsp')
, header = F
, title = "Preliminary Model - Covarience Check by Arm (Waves 1-2 only)"
, dep.var.caption = "Four Study Arms"
, dep.var.labels = c("UBI Ranking")

, column.labels = c("Pure Base", "Pure Ext", "Fair Base", "Fair Ext", "Control")
, notes = c("HC Robust Standard Errors"
, "Pure = Purity Frame | Fair = Fairness Frame"
, "Base = Base Only | Ext = Base + Extension")
, font.size = "small"
, column.sep.width = "1pt"
, label = "tab:covbalancecheck"
, type = stargazer_type
, single.row = TRUE
)

```
## Additional Regression Tables
```

{r model_arm_prelim, echo = FALSE, results='asis'}
prelim_model_libfair_prelim = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level, data =
results_armlibfair %>%
filter(recruitment_wave %in% c("Wave1", "Wave2")))
, clusters_in = NA)
prelim_model_libpure_prelim = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level, data =
results_armlibpure %>%
filter(recruitment_wave %in% c("Wave1", "Wave2")))
, clusters_in = NA)
prelim_model_confair_prelim = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level, data =
results_armconfair %>%
filter(recruitment_wave %in% c("Wave1", "Wave2")))
, clusters_in = NA)
prelim_model_conpure_prelim = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level, data =
results_armconpure %>%
filter(recruitment_wave %in% c("Wave1", "Wave2")))
, clusters_in = NA)

stargazer(prelim_model_libfair_prelim$lm, prelim_model_libpure_prelim$lm
, prelim_model_confair_prelim$lm, prelim_model_conpure_prelim$lm
, type = stargazer_type, header = F
, se = list(prelim_model_libfair_prelim$se_robust, prelim_model_libpure_prelim$se_robust
, prelim_model_confair_prelim$se_robust, prelim_model_conpure_prelim$se_robust)
, title = "Preliminary Model - By Arm (Waves 1-2 only)"
, column.labels = c("Lib + Fair", "Lib + Pure"

```
, "Con + Fair", "Con + Pure")
, covariate.labels = c("Base Treatment", "Extension Treatment")
, dep.var.caption = "Four Study Arms"
, dep.var.labels = "UBI Ranking"
, notes = "HC Robust Standard Errors"
, report = ('vcsp')
, font.size = "small"
, column.sep.width = "1pt"
, label = "tab:prelimmodel"
)
```

```
{r model_arm_recruitday, echo = FALSE, results='asis'}
model_control_day = custom_lm_calcs(lm_in = lm(ubi_number ~ ideology_bin +
recruitment_wave, data = results_clean_lim_ctrl), clusters_in = NA)

model_libfair_day = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level + recruitment_wave,
data = results_armlibfair), clusters_in = NA)
model_libpure_day = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level + recruitment_wave,
data = results_armlibpure), clusters_in = NA)
model_confair_day = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level + recruitment_wave,
data = results_armconfair), clusters_in = NA)
model_conpure_day = custom_lm_calcs(lm_in = lm(ubi_number ~ arm_level + recruitment_wave,
data = results_armconpure), clusters_in = NA)

stargazer(model_control_day$lm
, model_libfair_day$lm, model_libpure_day$lm
, model_confair_day$lm, model_conpure_day$lm
, type = stargazer_type, header = F
, se = list(model_control_day$se_robust
, model_libfair_day$se_robust, model_libpure_day$se_robust
, model_confair_day$se_robust, model_conpure_day$se_robust
)
, title = "By Arm, Recruitment Day Covariates"
, column.labels = c("Control Only"
, "Lib + Fair", "Lib + Pure"
, "Con + Fair", "Con + Pure")
, order = c(1,4,5,2,3,6,7)
```

```
, covariate.labels = c("Liberal", "Base Treatment", "Extension Treatment"
,"Wave 2", "Wave 3", "Wave 4", "Wave 5")
, dep.var.caption = "Four Study Arms + Control"
, dep.var.labels = "UBI Ranking"
, notes = "HC Robust Standard Errors"
, report = ('vcsp')
, font.size = "small"
, column.sep.width = "1pt"
, label = "tab:wavestratmodel"
)
```

\pagebreak

References