Final analysis code

Heather Kappes, Crystal Hall, Rebecca Johnson, Simone Zhang

01/06/2020

Table of Contents

# Load packages and setup

rm(list = ls())  
library(xlsx)  
library(dplyr)  
library(stringr)  
library(haven)  
library(ggplot2)  
library(tidyverse)  
library(data.table)  
library(viridis)  
library(kableExtra)  
library(qualtRics)  
library(car)  
library(blockrand)  
library(rtf)  
library(stargazer)  
library(xtable)  
  
theme\_new <- function(base\_size = 16, base\_family = "Helvetica"){  
 theme\_bw(base\_size = base\_size, base\_family = base\_family) %+replace%  
 theme(  
 panel.grid = element\_blank(),   
 panel.border = element\_rect(fill = NA, colour = "black", size=1),  
 panel.background = element\_rect(fill = "white", colour = "black"),   
 strip.background = element\_rect(fill = NA),  
 axis.text.x = element\_text(color = "black"),  
 axis.text.y = element\_text(color = "black")  
 )  
}  
  
  
code\_agg\_scales <- function(data, prefix\_string,  
 verbose = FALSE,  
 FUN){  
   
   
 ## get all columns with that prefix  
 cols\_withprefix = grep(sprintf("^%s", prefix\_string),  
 colnames(data),  
 value = TRUE)  
   
 if(verbose) print(sprintf("Coding scale based on: %s", paste(cols\_withprefix,   
 collapse = ";")))  
 ## apply coding function to all columns with that pattern  
 data[, cols\_withprefix] = apply(data[, cols\_withprefix], 2, FUN)  
   
 ## sum those columns and average by the number of columns considered  
 ### rj note-- right now, if respondent leaves an item blank (NA) it's  
 ### still summing the rest and just dividing by a different denom  
 ### can change that by doing na.rm = FALSE and changing   
 ### answer code- if respondent answers none, putting them as NA  
 sum\_across\_answered = rowSums(data[, cols\_withprefix], na.rm = TRUE)  
   
 ### get number of non-NA responses for that respondent for that scale  
 total\_items\_answered = rowSums(!is.na(data[, cols\_withprefix]))  
 avg\_across = ifelse(total\_items\_answered > 0,  
 (1/total\_items\_answered) \* sum\_across\_answered,  
 NA\_real\_)  
  
 # return the vector of averages   
 return(avg\_across)  
}

# Load and clean data

## Load and merge qualtrics to prolific attributes

# Load qualtrics data  
raw\_data\_init = read\_survey("../data/Beliefs about funding models \_finalv\_January 6, 2021\_17.21.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## StartDate = col\_datetime(format = ""),  
## EndDate = col\_datetime(format = ""),  
## Progress = col\_double(),  
## `Duration (in seconds)` = col\_double(),  
## Finished = col\_logical(),  
## RecordedDate = col\_datetime(format = ""),  
## RecipientLastName = col\_logical(),  
## RecipientFirstName = col\_logical(),  
## RecipientEmail = col\_logical(),  
## ExternalReference = col\_logical(),  
## LocationLatitude = col\_double(),  
## LocationLongitude = col\_double(),  
## Q\_RecaptchaScore = col\_double(),  
## w\_rankfair\_1 = col\_double(),  
## w\_rankfair\_2 = col\_double(),  
## w\_rankfair\_3 = col\_double(),  
## m\_rankfair\_1 = col\_double(),  
## m\_rankfair\_2 = col\_double(),  
## m\_rankfair\_3 = col\_double(),  
## `Create New Field or Choose From Dropdown...` = col\_logical()  
## )

## See spec(...) for full column specifications.

# From raw data, filter out ones that occurred pre-launch as part  
# of testing  
raw\_data <- raw\_data\_init %>%  
 filter(StartDate > as.Date("2020-12-27"))   
  
stopifnot(all(raw\_data$StartDate > as.Date("2020-12-27")))  
  
# Load prolific dem  
prol\_dem = read.csv("../data/prolific\_export\_5fd1326bfc1d530abecbcd16.csv")  
  
print(sprintf("Out of %s unique ids in prolific df, %s are found in qualtrics df",  
 length(unique(prol\_dem$participant\_id)),  
 length(intersect(unique(prol\_dem$participant\_id),  
 unique(raw\_data$prol\_id)))))

## [1] "Out of 1271 unique ids in prolific df, 1221 are found in qualtrics df"

## look at join errors of two types  
### people in prolific demographics but not found in qualtrics  
prol\_notfound = setdiff(prol\_dem$participant\_id,  
 raw\_data$prol\_id)  
  
### people in qualtrics but not found in prolific dem  
insurvey\_notprol = setdiff(raw\_data$prol\_id\_clean,  
 prol\_dem$participant\_id)  
  
### clean up prol id in qualtrics to try to match with prol dem  
raw\_data = raw\_data %>%  
 mutate(prol\_id\_clean =   
 gsub("^'-|@email.prolific.co|\\s+", "", tolower(prol\_id)))  
  
sprintf("Before cleaning, %s prol dem overlapped with qualtrics, after cleaning %s overlap",  
 length(unique(intersect(raw\_data$prol\_id, prol\_dem$participant\_id\_prolific))),  
 length(unique(intersect(raw\_data$prol\_id\_clean, prol\_dem$participant\_id\_prolific))))

## [1] "Before cleaning, 0 prol dem overlapped with qualtrics, after cleaning 0 overlap"

## rename prolific cols to distinguish b/t qualtrics cols  
colnames(prol\_dem) = sprintf("%s\_prolific",  
 colnames(prol\_dem))  
  
  
## rj note -- for now, just left as is and didnt try to fuzzy matching or str distance for entry errors  
## we could try to do  
   
## left join prolific cols onto raw data  
## so that all respondents in raw data are preserved regardless of prolific status  
raw\_data\_wp = merge(raw\_data,  
 prol\_dem,  
 by.x = "prol\_id\_clean",  
 by.y= "participant\_id\_prolific",  
 all.x = TRUE) %>%  
   
 # first, construct indicator for non-matches that are due to   
 # wrong prolific id length (should capture the testing, dartmouth, etc)  
 mutate(is\_prolid\_wronglength = ifelse(nchar(prol\_id\_clean) != 24, TRUE,   
 FALSE),  
 is\_failmerge\_prol = ifelse(!prol\_id\_clean %in% prol\_dem$participant\_id\_prolific,  
 TRUE, FALSE))  
  
  
### look at ones that fail merge-- see that they're either missing prolific  
### id entry, wrong length, testing/dartmouth, etc.- going to use wrong length  
### and fail merge later as exclusion criteria  
#View(raw\_data\_wp %>% filter(is\_failmerge\_prol))  
  
## clean up colnames  
## make colnames lowercase and remove spaces/punctuation  
colnames(raw\_data\_wp) = gsub("\\s+|\\(|\\)",   
 "",   
 tolower(colnames(raw\_data\_wp)))

## Clean and construct variables

## construct combined timing variables  
### get all time var (distributed based on what their ranking was)  
time\_var = grep("\\\_time$",  
 colnames(raw\_data\_wp),  
 value = TRUE)  
  
### code NA to blank  
raw\_data\_wp[, time\_var][is.na(raw\_data\_wp[,  
 time\_var])] <- ""  
raw\_data\_wp$timetrad\_raw = apply(raw\_data\_wp[, time\_var],  
 1,  
 function(x) paste(x, collapse = ""))   
  
## construct label for all choice combos  
fr\_cols = grep("fr\\\_1$", colnames(raw\_data\_wp),  
 value = TRUE)  
raw\_data\_wp$choice = gsub("\\\_fr\_1",  
 "",  
 names(raw\_data\_wp[, fr\_cols])[max.col(!is.na(raw\_data\_wp[, fr\_cols]), "first")])  
  
# Create flags and code vars  
## from looking at labels:  
### [w/m]\_rankfair\_1 == lottery  
### [w/m]\_rankfair\_2 == fcfs  
### [w/m]\_rankfair\_3 == points  
### and then values within are ranking  
raw\_data\_wp\_intermed = raw\_data\_wp %>%  
 ### renaming attention check item in jw scale  
 ### so that it's actually the attention and doesnt get aggregated  
 rename(attention\_check\_q = jw\_scale\_6,  
 political\_ideology = q34,  
 political\_affil= q35,  
 what\_researchers\_want = q37)  
  
raw\_data\_wp\_cleanvar = raw\_data\_wp\_intermed %>%  
 ### various screening  
 mutate(is\_failattention = case\_when(is.na(attention\_check\_q) ~ NA,  
 !grepl("Strong agreement", attention\_check\_q) ~ TRUE,   
 TRUE ~ FALSE),  
 is\_wrong\_distype = case\_when(distributionchannel != "anonymous" ~ TRUE,   
 TRUE ~ FALSE),  
 is\_cons\_revok = ifelse(grepl("REVOKE", sex\_prolific),  
 TRUE, FALSE),  
 is\_non100\_progress = ifelse(progress != 100, TRUE, FALSE),  
   
 ### for main tx, some are marked as finished == false and are missing condition  
 is\_any\_barriers = case\_when(cond %in% c("Wom", "Min") ~ TRUE,  
 cond == "No\_info" ~ FALSE,   
 TRUE ~ NA),  
   
 ## for rankings, combine ranking across randomization to  
 ## w or m points system (Wom cond = w, M cond = m,   
 ## no info cond = evenly split)  
 points\_rank= case\_when(  
 !is.na(w\_rankfair\_3) ~ w\_rankfair\_3,  
 !is.na(m\_rankfair\_3) ~ m\_rankfair\_3),  
 lottery\_rank = case\_when(  
 !is.na(w\_rankfair\_1) ~ w\_rankfair\_1,  
 !is.na(m\_rankfair\_1) ~ m\_rankfair\_1),  
 fcfs\_rank = case\_when(  
 !is.na(w\_rankfair\_2) ~ w\_rankfair\_2,  
 !is.na(m\_rankfair\_2) ~ m\_rankfair\_2),  
   
 ## additional screen  
 is\_missing\_DV = ifelse(is.na(points\_rank),  
 TRUE, FALSE),  
   
 ## reverse code points so that same direction as binary  
 ## where positive coef on treatment = more favorable  
 points\_rank\_rev = 3-points\_rank,  
 is\_points\_first = case\_when(points\_rank == 1 ~ TRUE,   
 points\_rank %in% c(2, 3) ~ FALSE,  
 TRUE ~ NA),  
   
 ## code categorical time into continuous # of weeks  
 timetrad\_weeks =   
 case\_when(grepl("as quickly", timetrad\_raw) ~ 0,  
 grepl("2 weeks", timetrad\_raw) ~ 2,  
 grepl("1 month", timetrad\_raw) ~ 4,  
 grepl("6 weeks", timetrad\_raw) ~ 6,  
 grepl("2 months", timetrad\_raw) ~ 8),  
   
 ## just world scale with non attention check items  
 ### rj note- as noted in function, if people skip some   
 ### items but complete others, coded using num/denom  
 ### with remaining items  
 ### gives warning since codes to NA for those missing all items  
 jw\_combined = code\_agg\_scales(raw\_data\_wp\_intermed,  
 prefix\_string = "jw\_scale",  
 FUN = function(x){  
 case\_when(grepl("disagreement", x) ~ 1,  
 grepl("agreement", x) ~ 6,  
 TRUE ~ as.numeric(x))}),  
   
 ### create numeric version of political ideology scale  
 ### with NA for "prefer not to say" and "not sure"  
 ### not using numeric analytically just for descriptive  
 ### higher = more conservative  
 political\_ideology\_numeric = case\_when(grepl("Very Conserv", political\_ideology) ~ 5,  
 grepl("Conserv", political\_ideology) ~ 4,  
 grepl("Mod", political\_ideology) ~ 3,  
 grepl("^Lib", political\_ideology) ~ 2,  
 grepl("Very Lib", political\_ideology) ~ 1,  
 TRUE ~ NA\_real\_),  
   
 ### various flags   
 is\_woman = case\_when(sex\_prolific == "Female" ~ TRUE,  
 sex\_prolific == "Male" ~ FALSE,   
 TRUE ~ NA),  
   
 ### primary definition of minority is non-white  
 is\_minority = case\_when(is.na(raceeth) ~ NA,  
 !raceeth %in% c("White", "Prefer not to say", "White, Prefer not to say") ~ TRUE,   
 TRUE ~ FALSE),  
   
 ### secondary definition is only hispanic or black  
 ### for both can select multiple  
 is\_minority\_sec = case\_when(is.na(raceeth) ~ NA,  
 grepl("Hispanic|Black", raceeth) ~ TRUE,   
 TRUE ~ FALSE),  
   
 ### group into broader categories  
 ### mainly comparing D, R, I  
 political\_affil\_buckets = case\_when(  
 political\_affil %in% c("No preference",  
 "Other party",  
 "Prefer not to say") |  
 is.na(political\_affil) ~ "Other", # include skips in others   
 TRUE ~ political\_affil))

## Warning: Problem with `mutate()` input `jw\_combined`.  
## ℹ NAs introduced by coercion  
## ℹ Input `jw\_combined` is `code\_agg\_scales(...)`.  
  
## Warning: Problem with `mutate()` input `jw\_combined`.  
## ℹ NAs introduced by coercion  
## ℹ Input `jw\_combined` is `code\_agg\_scales(...)`.  
  
## Warning: Problem with `mutate()` input `jw\_combined`.  
## ℹ NAs introduced by coercion  
## ℹ Input `jw\_combined` is `code\_agg\_scales(...)`.  
  
## Warning: Problem with `mutate()` input `jw\_combined`.  
## ℹ NAs introduced by coercion  
## ℹ Input `jw\_combined` is `code\_agg\_scales(...)`.  
  
## Warning: Problem with `mutate()` input `jw\_combined`.  
## ℹ NAs introduced by coercion  
## ℹ Input `jw\_combined` is `code\_agg\_scales(...)`.  
  
## Warning: Problem with `mutate()` input `jw\_combined`.  
## ℹ NAs introduced by coercion  
## ℹ Input `jw\_combined` is `code\_agg\_scales(...)`.  
  
## Warning: Problem with `mutate()` input `jw\_combined`.  
## ℹ NAs introduced by coercion  
## ℹ Input `jw\_combined` is `code\_agg\_scales(...)`.

## Load free responses answers and reviewer-confirmed nonsensical

fr\_coded\_z = read.csv("../data/frcoding\_finalsurvey - Xinzhe.csv")  
fr\_coded\_k = read.csv("../data/frcoding\_finalsurvey - K to code.csv") %>%  
 rename(`Considers.relevant.appropriate.factors` =   
 `Considers.relevant.appropriate.factors...the.preferred.method.DOES.this..and.or.the.method.they.don.t.like.does.NOT.do.this.`)  
  
fr\_coded\_both = rbind.data.frame(fr\_coded\_z,  
 fr\_coded\_k)  
  
## read in final nonsensical codes based on review  
nonsense\_codes = read.csv("../data/frcoding\_finalsurvey - flag\_nonsensical.csv")  
  
### get prolific ids of ones flagged as yes nonsensical reviewer agree  
nonsens\_prol = nonsense\_codes %>%  
 filter(`reviewer\_agree..1...yes..0...no.` == 1) %>%  
 pull(prol\_id)  
  
## create flag in analytic df and exclude from analyses  
raw\_data\_wp\_cleanvar = raw\_data\_wp\_cleanvar %>%  
 mutate(is\_nonsens\_answer = ifelse(prol\_id\_clean %in% nonsens\_prol, TRUE, FALSE))  
  
## some checks  
stopifnot(sum(raw\_data\_wp\_cleanvar$is\_points\_first,  
 na.rm = TRUE) ==   
 nrow(raw\_data\_wp\_cleanvar %>% filter(points\_rank == 1)))  
  
# Identify any additional exact duplicates   
exact\_duplicates = fr\_coded\_both %>%  
 group\_by(fr\_why\_first\_over\_second) %>%  
 filter(n() > 1)  
  
# 3 additional observations that are duplicates that were not flagged by RA review  
exact\_duplicates$prol\_id[!exact\_duplicates$prol\_id %in% nonsens\_prol]

## [1] 5fd3e04162a6810ed9ab4041 5fd63da4d997394f28dfdd2e  
## [3] 5fe273c2247470e5d698e061  
## 1128 Levels: 546e3778fdf99b2bc7ebcff6 ... 5feaa1772178d1ae84391de6

There were 3 additional observations that were duplicates that were not flagged by RA review.

# Summarize descriptive stats

Use two versions of the data

1. Same version as above
2. model\_df: meets all inclusion criteria

## Randomization checks (just with main data; not analytic)

# Distribution of duration across conditions  
## median duration is 6 minutes  
## slightly lower for the people who read about historical barriers  
quantile(raw\_data\_wp\_cleanvar$durationinseconds)

## 0% 25% 50% 75% 100%   
## 7.0 326.5 463.0 681.5 7565.0

raw\_data\_wp\_cleanvar %>%  
 group\_by(is\_any\_barriers) %>%  
 summarise(quant = paste(quantile(durationinseconds),  
 collapse = "; "))

## # A tibble: 3 x 2  
## is\_any\_barriers quant   
## \* <lgl> <chr>   
## 1 FALSE 97; 315; 452; 662.5; 3418  
## 2 TRUE 82; 338; 469; 694; 7565   
## 3 NA 7; 41; 107; 733; 4253

# N per condition- nas still from noncompleters  
table(raw\_data\_wp\_cleanvar$cond, useNA = "always")

##   
## Min No\_info Wom <NA>   
## 410 411 411 15

# Check that free response options were displayed correctly  
rank\_cols = grep("\\\_rank$", colnames(raw\_data\_wp\_cleanvar),  
 value = TRUE)   
check\_logic <- function(one\_fr){  
   
 ## first filter to those who filled out the fr  
 fill\_resp = raw\_data\_wp\_cleanvar %>%  
 filter(!is.na(!!sym(one\_fr)))  
   
 ## then, what the first choice should be  
 choices = unlist(strsplit(gsub("\\\_fr\\\_1", "", one\_fr),  
 split = ""))  
 first = choices[1]  
 second = choices[2]  
   
 ## make sure rank matches choices  
 ### first   
 if((first == "p" & all(fill\_resp$points\_rank == 1)) |  
 (first == "f" & all(fill\_resp$fcfs\_rank == 1)) |  
 (first == "l" & all(fill\_resp$lottery\_rank == 1))){  
 print(sprintf("passed first choice for: %s",  
 one\_fr))  
 } else{  
 print(sprintf("failed first choice for: %s",  
 one\_fr))  
 }  
   
 ### second  
 if((second == "p" & all(fill\_resp$points\_rank == 2)) |  
 (second == "f" & all(fill\_resp$fcfs\_rank == 2)) |  
 (second == "l" & all(fill\_resp$lottery\_rank == 2))){  
 print(sprintf("passed second choice for: %s",  
 one\_fr))  
 } else{  
 print(sprintf("failed second choice for: %s",  
 one\_fr))  
 }  
 return(NULL)  
   
}  
  
checking <- lapply(fr\_cols, check\_logic)

## [1] "passed first choice for: pf\_fr\_1"  
## [1] "passed second choice for: pf\_fr\_1"  
## [1] "passed first choice for: pl\_fr\_1"  
## [1] "passed second choice for: pl\_fr\_1"  
## [1] "passed first choice for: lp\_fr\_1"  
## [1] "passed second choice for: lp\_fr\_1"  
## [1] "passed first choice for: lf\_fr\_1"  
## [1] "passed second choice for: lf\_fr\_1"  
## [1] "passed first choice for: fl\_fr\_1"  
## [1] "passed second choice for: fl\_fr\_1"  
## [1] "passed first choice for: fp\_fr\_1"  
## [1] "passed second choice for: fp\_fr\_1"

## Demographics/attitudes

# Demographics  
dem\_vars = c("sex\_prolific",  
 "raceeth",  
 "political\_affil",  
 "political\_ideology",  
 "political\_ideology\_numeric",  
 "is\_woman",  
 "is\_minority",  
 "what\_researchers\_want",  
 "political\_affil\_buckets")  
  
## filter to modeling df  
## that passes checks:  
## (1) sensical FR,  
## (2) non-missing DV  
## (3) didnt revoke consent  
## (4) passes attention check embedded in JW scale  
## (5) comes from correct distribution channel and   
## (6) linkable to their prolific attributes (so entered  
## code correctly)  
model\_df = raw\_data\_wp\_cleanvar %>%  
 filter(!is\_nonsens\_answer & !is\_cons\_revok &  
 !is\_missing\_DV &  
 (!(is\_failattention | is.na(is\_failattention))) &  
 !is\_wrong\_distype &  
 !is\_failmerge\_prol &  
 (!(is\_prolid\_wronglength | is.na(is\_prolid\_wronglength))) &  
 !is\_non100\_progress)  
  
## print n fail each filter (n filtered out is < sum due to overlap)  
lapply(raw\_data\_wp\_cleanvar[, c("is\_nonsens\_answer",  
 "is\_cons\_revok",  
 "is\_missing\_DV",  
 "is\_failattention",  
 "is\_wrong\_distype",  
 "is\_failmerge\_prol",  
 "is\_prolid\_wronglength",  
 "is\_non100\_progress")],  
 function(x) table(x, useNA = "always"))

## $is\_nonsens\_answer  
## x  
## FALSE TRUE <NA>   
## 1220 27 0   
##   
## $is\_cons\_revok  
## x  
## FALSE TRUE <NA>   
## 1222 25 0   
##   
## $is\_missing\_DV  
## x  
## FALSE TRUE <NA>   
## 1147 100 0   
##   
## $is\_failattention  
## x  
## FALSE TRUE <NA>   
## 1178 52 17   
##   
## $is\_wrong\_distype  
## x  
## FALSE TRUE <NA>   
## 1246 1 0   
##   
## $is\_failmerge\_prol  
## x  
## FALSE TRUE <NA>   
## 1225 22 0   
##   
## $is\_prolid\_wronglength  
## x  
## FALSE TRUE <NA>   
## 1229 9 9   
##   
## $is\_non100\_progress  
## x  
## FALSE TRUE <NA>   
## 1210 37 0

sprintf("Once we filter, goes from %s to %s",  
 nrow(raw\_data\_wp\_cleanvar),  
 nrow(model\_df))

## [1] "Once we filter, goes from 1247 to 1052"

## dem breakdowns pre filter  
lapply(raw\_data\_wp\_cleanvar[, dem\_vars],  
 function(x) table(x))

## $sex\_prolific  
## x  
## CONSENT REVOKED Female Male   
## 25 614 586   
##   
## $raceeth  
## x  
## American Indian or Alaska Native   
## 3   
## Asian   
## 71   
## Black or African American   
## 154   
## Black or African American,Asian,American Indian or Alaska Native   
## 1   
## Black or African American,Asian,Native Hawaiian or Pacific Islander   
## 1   
## Hispanic, Latino, or Spanish   
## 40   
## Hispanic, Latino, or Spanish,Asian   
## 2   
## Hispanic, Latino, or Spanish,Black or African American   
## 4   
## Hispanic, Latino, or Spanish,Black or African American,American Indian or Alaska Native   
## 1   
## Middle Eastern or North African   
## 2   
## Prefer not to say   
## 5   
## Some other race, ethnicity, or origin   
## 3   
## White   
## 881   
## White,American Indian or Alaska Native   
## 8   
## White,Asian   
## 11   
## White,Black or African American   
## 4   
## White,Hispanic, Latino, or Spanish   
## 32   
## White,Hispanic, Latino, or Spanish,Black or African American   
## 1   
## White,Middle Eastern or North African   
## 4   
## White,Prefer not to say   
## 2   
## White,Some other race, ethnicity, or origin   
## 2   
##   
## $political\_affil  
## x  
## Democrat Independent No preference Other party   
## 611 282 51 23   
## Prefer not to say Republican   
## 4 257   
##   
## $political\_ideology  
## x  
## Conservative Liberal Moderate Not Sure   
## 206 407 303 13   
## Prefer not to say Very Conservative Very Liberal   
## 9 99 195   
##   
## $political\_ideology\_numeric  
## x  
## 1 2 3 4 5   
## 195 407 303 206 99   
##   
## $is\_woman  
## x  
## FALSE TRUE   
## 586 614   
##   
## $is\_minority  
## x  
## FALSE TRUE   
## 886 346   
##   
## $what\_researchers\_want  
## x  
## I don't think the researcher cared which method I said was the most fair.   
## 675   
## The researcher definitely wanted me to say first-come first-served was the most fair.   
## 29   
## The researcher definitely wanted me to say lottery was the most fair.   
## 37   
## The researcher definitely wanted me to say points was the most fair.   
## 95   
## The researcher might have wanted me to say first-come first-served was the most fair.   
## 39   
## The researcher might have wanted me to say lottery was the most fair.   
## 45   
## The researcher might have wanted me to say points was the most fair.   
## 165   
## The researcher probably wanted me to say one method was the most fair, but I don't know which method that was.   
## 125   
##   
## $political\_affil\_buckets  
## x  
## Democrat Independent Other Republican   
## 611 282 97 257

## dem breakdowns post filter  
lapply(model\_df[, dem\_vars],  
 function(x) table(x))

## $sex\_prolific  
## x  
## CONSENT REVOKED Female Male   
## 0 554 498   
##   
## $raceeth  
## x  
## American Indian or Alaska Native   
## 2   
## Asian   
## 63   
## Black or African American   
## 134   
## Black or African American,Asian,American Indian or Alaska Native   
## 1   
## Hispanic, Latino, or Spanish   
## 31   
## Hispanic, Latino, or Spanish,Asian   
## 2   
## Hispanic, Latino, or Spanish,Black or African American   
## 4   
## Hispanic, Latino, or Spanish,Black or African American,American Indian or Alaska Native   
## 1   
## Middle Eastern or North African   
## 2   
## Prefer not to say   
## 4   
## Some other race, ethnicity, or origin   
## 2   
## White   
## 746   
## White,American Indian or Alaska Native   
## 7   
## White,Asian   
## 11   
## White,Black or African American   
## 4   
## White,Hispanic, Latino, or Spanish   
## 30   
## White,Hispanic, Latino, or Spanish,Black or African American   
## 1   
## White,Middle Eastern or North African   
## 4   
## White,Prefer not to say   
## 1   
## White,Some other race, ethnicity, or origin   
## 2   
##   
## $political\_affil  
## x  
## Democrat Independent No preference Other party   
## 525 240 42 23   
## Prefer not to say Republican   
## 2 216   
##   
## $political\_ideology  
## x  
## Conservative Liberal Moderate Not Sure   
## 169 365 260 11   
## Prefer not to say Very Conservative Very Liberal   
## 6 70 171   
##   
## $political\_ideology\_numeric  
## x  
## 1 2 3 4 5   
## 171 365 260 169 70   
##   
## $is\_woman  
## x  
## FALSE TRUE   
## 498 554   
##   
## $is\_minority  
## x  
## FALSE TRUE   
## 750 302   
##   
## $what\_researchers\_want  
## x  
## I don't think the researcher cared which method I said was the most fair.   
## 605   
## The researcher definitely wanted me to say first-come first-served was the most fair.   
## 19   
## The researcher definitely wanted me to say lottery was the most fair.   
## 21   
## The researcher definitely wanted me to say points was the most fair.   
## 73   
## The researcher might have wanted me to say first-come first-served was the most fair.   
## 34   
## The researcher might have wanted me to say lottery was the most fair.   
## 40   
## The researcher might have wanted me to say points was the most fair.   
## 144   
## The researcher probably wanted me to say one method was the most fair, but I don't know which method that was.   
## 116   
##   
## $political\_affil\_buckets  
## x  
## Democrat Independent Other Republican   
## 525 240 71 216

lapply(model\_df[, dem\_vars],  
 function(x) prop.table(table(x)))

## $sex\_prolific  
## x  
## CONSENT REVOKED Female Male   
## 0.000000 0.526616 0.473384   
##   
## $raceeth  
## x  
## American Indian or Alaska Native   
## 0.0019011407   
## Asian   
## 0.0598859316   
## Black or African American   
## 0.1273764259   
## Black or African American,Asian,American Indian or Alaska Native   
## 0.0009505703   
## Hispanic, Latino, or Spanish   
## 0.0294676806   
## Hispanic, Latino, or Spanish,Asian   
## 0.0019011407   
## Hispanic, Latino, or Spanish,Black or African American   
## 0.0038022814   
## Hispanic, Latino, or Spanish,Black or African American,American Indian or Alaska Native   
## 0.0009505703   
## Middle Eastern or North African   
## 0.0019011407   
## Prefer not to say   
## 0.0038022814   
## Some other race, ethnicity, or origin   
## 0.0019011407   
## White   
## 0.7091254753   
## White,American Indian or Alaska Native   
## 0.0066539924   
## White,Asian   
## 0.0104562738   
## White,Black or African American   
## 0.0038022814   
## White,Hispanic, Latino, or Spanish   
## 0.0285171103   
## White,Hispanic, Latino, or Spanish,Black or African American   
## 0.0009505703   
## White,Middle Eastern or North African   
## 0.0038022814   
## White,Prefer not to say   
## 0.0009505703   
## White,Some other race, ethnicity, or origin   
## 0.0019011407   
##   
## $political\_affil  
## x  
## Democrat Independent No preference Other party   
## 0.500954198 0.229007634 0.040076336 0.021946565   
## Prefer not to say Republican   
## 0.001908397 0.206106870   
##   
## $political\_ideology  
## x  
## Conservative Liberal Moderate Not Sure   
## 0.160646388 0.346958175 0.247148289 0.010456274   
## Prefer not to say Very Conservative Very Liberal   
## 0.005703422 0.066539924 0.162547529   
##   
## $political\_ideology\_numeric  
## x  
## 1 2 3 4 5   
## 0.16521739 0.35265700 0.25120773 0.16328502 0.06763285   
##   
## $is\_woman  
## x  
## FALSE TRUE   
## 0.473384 0.526616   
##   
## $is\_minority  
## x  
## FALSE TRUE   
## 0.7129278 0.2870722   
##   
## $what\_researchers\_want  
## x  
## I don't think the researcher cared which method I said was the most fair.   
## 0.57509506   
## The researcher definitely wanted me to say first-come first-served was the most fair.   
## 0.01806084   
## The researcher definitely wanted me to say lottery was the most fair.   
## 0.01996198   
## The researcher definitely wanted me to say points was the most fair.   
## 0.06939163   
## The researcher might have wanted me to say first-come first-served was the most fair.   
## 0.03231939   
## The researcher might have wanted me to say lottery was the most fair.   
## 0.03802281   
## The researcher might have wanted me to say points was the most fair.   
## 0.13688213   
## The researcher probably wanted me to say one method was the most fair, but I don't know which method that was.   
## 0.11026616   
##   
## $political\_affil\_buckets  
## x  
## Democrat Independent Other Republican   
## 0.49904943 0.22813688 0.06749049 0.20532319

## n per conditions  
table(model\_df$cond) ### still about even, slightly lower in minorities

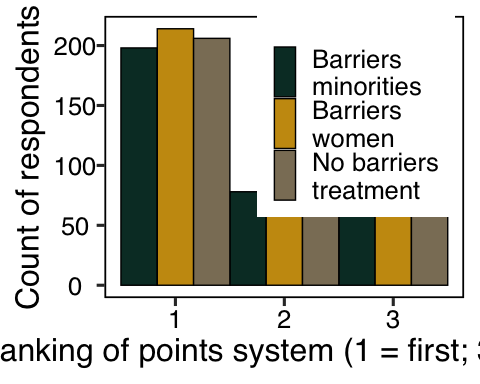
##   
## Min No\_info Wom   
## 340 352 360

table(model\_df$is\_any\_barriers)

##   
## FALSE TRUE   
## 352 700

# Descriptives on outcome

# add descriptive condition  
model\_df = model\_df %>%  
 mutate(cond\_4graph = case\_when(cond == "Min" ~ "Barriers\nminorities",  
 cond == "Wom" ~ "Barriers\nwomen",  
 cond == "No\_info" ~ "No barriers\ntreatment"),  
 barriers\_4graph = case\_when(is\_any\_barriers ~ "Either barriers\ntreatment",  
 TRUE ~ "No barriers\ntreatment"))  
  
## color map - can change if we change  
# https://www.color-hex.com/color-palette/103796  
col\_map\_detailed = c("Barriers\nminorities" = "#09382f",  
 "Barriers\nwomen" = "#c99910",  
 "No barriers\ntreatment" = "wheat4")  
col\_map\_coarse = c("Either barriers\ntreatment" = "#be3d3d",  
 "No barriers\ntreatment" = "wheat4")  
  
## first, show raw distribution of ranks across three groups  
dist\_rank\_plot <- ggplot(model\_df, aes(x = points\_rank,  
 group = cond\_4graph,  
 fill = cond\_4graph)) +  
 geom\_histogram(bins = 3, position = "dodge", color = "black") +  
 theme\_new(base\_size = 24) +  
 xlab("Ranking of points system (1 = first; 3 = last)") +  
 theme(legend.position = c(0.7, 0.7)) +  
 labs(fill = "") +  
 ylab("Count of respondents") +  
 scale\_fill\_manual(values = col\_map\_detailed)   
  
dist\_rank\_plot

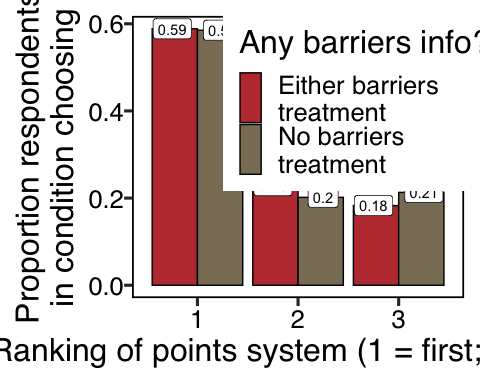


ggsave("../output/jap\_paper/figs/rankings\_count\_3groups.png",  
 plot = dist\_rank\_plot,  
 width = 12,  
 height = 8)   
  
  
## proportions choose: barriers versus not  
## since uneven counts between any barriers and   
## no barriers, get proportions  
prop\_choose = model\_df %>%  
 group\_by(points\_rank, barriers\_4graph) %>%  
 summarise(num = n()) %>%  
 left\_join(model\_df %>%  
 group\_by(barriers\_4graph) %>%  
 summarise(denom = n())) %>%  
 mutate(prop\_choose = num/denom) %>%  
 ungroup()

## `summarise()` has grouped output by 'points\_rank'. You can override using the `.groups` argument.

## Joining, by = "barriers\_4graph"

prop\_choose\_plot <- ggplot(prop\_choose, aes(x = points\_rank,  
 y = prop\_choose,  
 group = barriers\_4graph,  
 fill = barriers\_4graph)) +  
 geom\_bar(stat = "identity",  
 position = "dodge", col = "black") +  
 theme\_new(base\_size = 24) +  
 scale\_fill\_manual(values = col\_map\_coarse) +  
 xlab("Ranking of points system (1 = first; 3 = last)") +  
 theme(legend.position = c(0.7, 0.7)) +  
 labs(fill = "Any barriers info?") +  
 geom\_label(aes(x = points\_rank, y = prop\_choose,  
 group = barriers\_4graph,  
  
 label = round(prop\_choose, 2)),  
 position = position\_dodge(width = 1),  
 fill = "white") +  
 ylab("Proportion respondents\nin condition choosing")  
  
prop\_choose\_plot

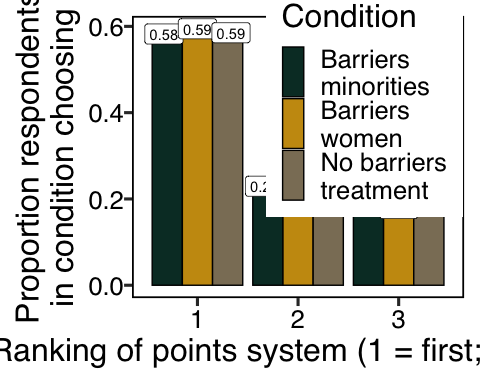


ggsave("../output/jap\_paper/figs/rankings\_prop\_2groups.png",  
 plot = prop\_choose\_plot,  
 width = 12,  
 height = 8)   
  
  
## repeat but for the more detailed barriers condition  
prop\_choose\_detail = model\_df %>%  
 group\_by(points\_rank, cond\_4graph) %>%  
 summarise(num = n()) %>%  
 left\_join(model\_df %>%  
 group\_by(cond\_4graph) %>%  
 summarise(denom = n())) %>%  
 mutate(prop\_choose = num/denom) %>%  
 ungroup()

## `summarise()` has grouped output by 'points\_rank'. You can override using the `.groups` argument.

## Joining, by = "cond\_4graph"

prop\_choose\_plot\_detail <- ggplot(prop\_choose\_detail, aes(x = points\_rank,  
 y = prop\_choose,  
 group = cond\_4graph,  
 fill = cond\_4graph)) +  
 geom\_bar(stat = "identity",  
 position = "dodge", col = "black") +  
 theme\_new(base\_size = 24) +  
 scale\_fill\_manual(values = col\_map\_detailed) +  
 xlab("Ranking of points system (1 = first; 3 = last)") +  
 theme(legend.position = c(0.7, 0.7)) +  
 labs(fill = "Condition") +  
 geom\_label(aes(x = points\_rank, y = prop\_choose,  
 group = cond\_4graph,  
  
 label = round(prop\_choose, 2)),  
 position = position\_dodge(width = 1),  
 fill = "white") +  
 ylab("Proportion respondents\nin condition choosing")  
  
prop\_choose\_plot\_detail



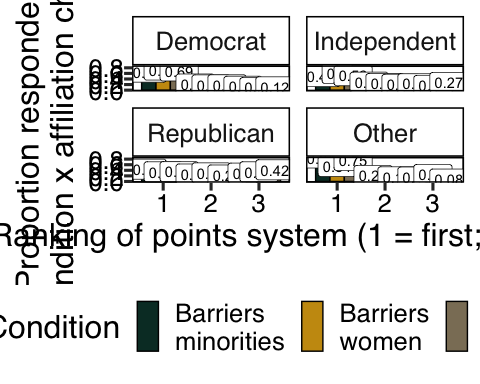
ggsave("../output/jap\_paper/figs/rankings\_prop\_3groups.png",  
 plot = prop\_choose\_plot\_detail,  
 width = 12,  
 height = 8)   
  
## do figure on raw rates by political affiliation  
prop\_choose\_detail\_waffil = model\_df %>%  
 group\_by(points\_rank, cond\_4graph, political\_affil\_buckets) %>%  
 summarise(num = n()) %>%  
 left\_join(model\_df %>%  
 group\_by(cond\_4graph, political\_affil\_buckets) %>%  
 summarise(denom = n())) %>%  
 mutate(prop\_choose = num/denom) %>%  
 ungroup()

## `summarise()` has grouped output by 'points\_rank', 'cond\_4graph'. You can override using the `.groups` argument.

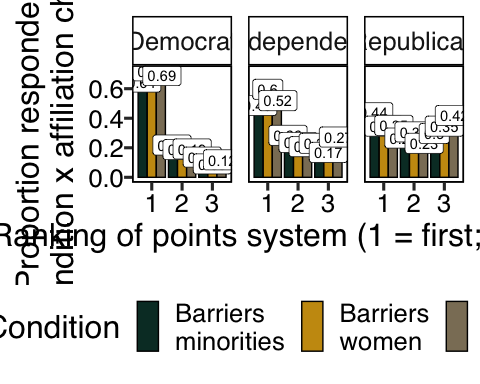
## `summarise()` has grouped output by 'cond\_4graph'. You can override using the `.groups` argument.

## Joining, by = c("cond\_4graph", "political\_affil\_buckets")

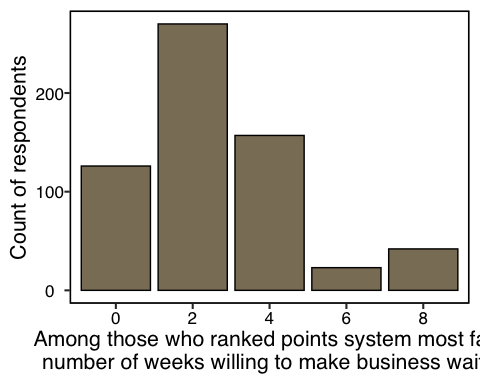
prop\_choose\_plot\_detail\_waffil <- ggplot(prop\_choose\_detail\_waffil, aes(x = points\_rank,  
 y = prop\_choose,  
 group = cond\_4graph,  
 fill = cond\_4graph)) +  
 geom\_bar(stat = "identity",  
 position = "dodge", col = "black") +  
 theme\_new(base\_size = 24) +  
 scale\_fill\_manual(values = col\_map\_detailed) +  
 xlab("Ranking of points system (1 = first; 3 = last)") +  
 theme(legend.position = "bottom") +  
 labs(fill = "Condition") +  
 geom\_label(aes(x = points\_rank, y = prop\_choose,  
 group = cond\_4graph,  
  
 label = round(prop\_choose, 2)),  
 position = position\_dodge(width = 1),  
 fill = "white") +  
 ylab("Proportion respondents\nin condition x affiliation choosing") +  
 facet\_wrap(~factor(political\_affil\_buckets,  
 levels = c("Democrat", "Independent",  
 "Republican", "Other"),  
 ordered = TRUE), ncol = 2)  
  
prop\_choose\_plot\_detail\_waffil



ggsave("../output/jap\_paper/figs/rankings\_prop\_3groups\_wpoliticalaffil.png",  
 plot = prop\_choose\_plot\_detail\_waffil,  
 width = 12,  
 height = 8)   
  
prop\_choose\_plot\_detail\_waffil\_excludeother <- ggplot(prop\_choose\_detail\_waffil %>%  
 filter(political\_affil\_buckets != "Other"), aes(x = points\_rank,  
 y = prop\_choose,  
 group = cond\_4graph,  
 fill = cond\_4graph)) +  
 geom\_bar(stat = "identity",  
 position = "dodge", col = "black") +  
 theme\_new(base\_size = 24) +  
 scale\_fill\_manual(values = col\_map\_detailed) +  
 xlab("Ranking of points system (1 = first; 3 = last)") +  
 theme(legend.position = "bottom") +  
 labs(fill = "Condition") +  
 geom\_label(aes(x = points\_rank, y = prop\_choose,  
 group = cond\_4graph,  
  
 label = round(prop\_choose, 2)),  
 position = position\_dodge(width = 1),  
 fill = "white") +  
 ylab("Proportion respondents\nin condition x affiliation choosing") +  
 facet\_wrap(~factor(political\_affil\_buckets,  
 levels = c("Democrat", "Independent",  
 "Republican"),  
 ordered = TRUE), ncol = 3)  
  
prop\_choose\_plot\_detail\_waffil\_excludeother



ggsave("../output/jap\_paper/figs/rankings\_prop\_3groups\_wpoliticalaffil\_excludeother.png",  
 plot = prop\_choose\_plot\_detail\_waffil\_excludeother,  
 width = 12,  
 height = 8)   
  
## For those who chose points, how many weeks are they willing to wait  
## one missing  
time\_plot <- ggplot(model\_df %>%  
 filter(is\_points\_first &  
 !is.na(timetrad\_weeks)), aes(x = factor(timetrad\_weeks))) +  
 geom\_bar(stat = "count", fill = "wheat4", col = "black") +  
 xlab("Among those who ranked points system most fair,\nnumber of weeks willing to make business wait?") +  
 theme\_new() +  
 ylab("Count of respondents")  
  
time\_plot



ggsave("../output/jap\_paper/figs/among\_points\_timedist.png",  
 plot = time\_plot,  
 width = 12,  
 height = 8)

# Analytic

## Hyp 1: barriers tx causes (1) higher ranking of points system (lower rank) and (2) higher likelihood of ranking points first

## continuous regressing rank of points  
## reverse coded so that pos coef = rank higher  
summary(lm(points\_rank\_rev ~ is\_any\_barriers,  
 data = model\_df))

##   
## Call:  
## lm(formula = points\_rank\_rev ~ is\_any\_barriers, data = model\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.4057 -0.4057 0.5943 0.5943 0.6278   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.37216 0.04216 32.545 <2e-16 \*\*\*  
## is\_any\_barriersTRUE 0.03356 0.05169 0.649 0.516   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.791 on 1050 degrees of freedom  
## Multiple R-squared: 0.0004012, Adjusted R-squared: -0.0005508   
## F-statistic: 0.4215 on 1 and 1050 DF, p-value: 0.5164

## binary regressing points as first  
## use lpm  
summary(lm(is\_points\_first ~ is\_any\_barriers,  
 data = model\_df))

##   
## Call:  
## lm(formula = is\_points\_first ~ is\_any\_barriers, data = model\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5886 -0.5886 0.4114 0.4114 0.4148   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.585227 0.026264 22.282 <2e-16 \*\*\*  
## is\_any\_barriersTRUE 0.003344 0.032197 0.104 0.917   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4928 on 1050 degrees of freedom  
## Multiple R-squared: 1.027e-05, Adjusted R-squared: -0.0009421   
## F-statistic: 0.01079 on 1 and 1050 DF, p-value: 0.9173

## save  
stargazer(lm(points\_rank\_rev ~ is\_any\_barriers,  
 data = model\_df),  
 lm(is\_points\_first ~ is\_any\_barriers,  
 data = model\_df),  
 type = "html",  
 out="../output/jap\_paper/tables/maineffect\_regs.doc")

##   
## <table style="text-align:center"><tr><td colspan="3" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"></td><td colspan="2"><em>Dependent variable:</em></td></tr>  
## <tr><td></td><td colspan="2" style="border-bottom: 1px solid black"></td></tr>  
## <tr><td style="text-align:left"></td><td>points\_rank\_rev</td><td>is\_points\_first</td></tr>  
## <tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td></tr>  
## <tr><td colspan="3" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">is\_any\_barriers</td><td>0.034</td><td>0.003</td></tr>  
## <tr><td style="text-align:left"></td><td>(0.052)</td><td>(0.032)</td></tr>  
## <tr><td style="text-align:left"></td><td></td><td></td></tr>  
## <tr><td style="text-align:left">Constant</td><td>1.372<sup>\*\*\*</sup></td><td>0.585<sup>\*\*\*</sup></td></tr>  
## <tr><td style="text-align:left"></td><td>(0.042)</td><td>(0.026)</td></tr>  
## <tr><td style="text-align:left"></td><td></td><td></td></tr>  
## <tr><td colspan="3" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">Observations</td><td>1,052</td><td>1,052</td></tr>  
## <tr><td style="text-align:left">R<sup>2</sup></td><td>0.0004</td><td>0.00001</td></tr>  
## <tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>-0.001</td><td>-0.001</td></tr>  
## <tr><td style="text-align:left">Residual Std. Error (df = 1050)</td><td>0.791</td><td>0.493</td></tr>  
## <tr><td style="text-align:left">F Statistic (df = 1; 1050)</td><td>0.421</td><td>0.011</td></tr>  
## <tr><td colspan="3" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"><em>Note:</em></td><td colspan="2" style="text-align:right"><sup>\*</sup>p<0.1; <sup>\*\*</sup>p<0.05; <sup>\*\*\*</sup>p<0.01</td></tr>  
## </table>

## Hyp 2: barriers tx causes respondents to rate it more important to prioritize that group

RJ note: programming issue so cant use (or can only use for female condition) since issue w/ parallel q structure across W and M

## notes on importance ratings  
# The first asks respondents whether it is important for a method toprioritize [women/minority-owned] businesses (Yes; No; Not sure). The secondquestion depends on responses to the first question: respondents who indicatedthat it wasnotimportant to prioritize [women/minority-owned] businesses will beasked how important it is that the underserved group was not advantaged, whilerespondents who indicated that itwasimportant to prioritize those businesses willbe asked how important it is that the underserved group got an advantage. Wewill combine these two questions to create a scale ranging from -3 (very important5  
# that underserved group got no advantage) to 0 (not sure) to 3 (very importantthat underserved group got an advantage)

## Hyp 3: differences in tx effect between minority barriers treatment and women’s barrier treatment

1. Filter to respondents randomized to those two
2. Reg is comparing those two

## filter to some barrier randomization  
barriers\_r = model\_df %>%  
 filter(cond != "No\_info")   
  
  
## when women is the tx, slightly more likely  
## to rank points higher/first than when minority is tx  
summary(lm(points\_rank\_rev ~ cond,  
 data = barriers\_r))

##   
## Call:  
## lm(formula = points\_rank\_rev ~ cond, data = barriers\_r)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.4167 -0.4167 0.5833 0.6059 0.6059   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.39412 0.04230 32.956 <2e-16 \*\*\*  
## condWom 0.02255 0.05899 0.382 0.702   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.78 on 698 degrees of freedom  
## Multiple R-squared: 0.0002093, Adjusted R-squared: -0.001223   
## F-statistic: 0.1461 on 1 and 698 DF, p-value: 0.7024

summary(lm(is\_points\_first ~ cond,  
 data = barriers\_r))

##   
## Call:  
## lm(formula = is\_points\_first ~ cond, data = barriers\_r)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5944 -0.5824 0.4056 0.4177 0.4177   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.58235 0.02672 21.792 <2e-16 \*\*\*  
## condWom 0.01209 0.03726 0.324 0.746   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4928 on 698 degrees of freedom  
## Multiple R-squared: 0.0001508, Adjusted R-squared: -0.001282   
## F-statistic: 0.1053 on 1 and 698 DF, p-value: 0.7457

## save  
stargazer(lm(points\_rank\_rev ~ cond,  
 data = barriers\_r),  
 lm(is\_points\_first ~ cond,  
 data = barriers\_r),  
 type = "html",  
 out="../output/jap\_paper/tables/womenvmob\_barriers\_regs.doc")

##   
## <table style="text-align:center"><tr><td colspan="3" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"></td><td colspan="2"><em>Dependent variable:</em></td></tr>  
## <tr><td></td><td colspan="2" style="border-bottom: 1px solid black"></td></tr>  
## <tr><td style="text-align:left"></td><td>points\_rank\_rev</td><td>is\_points\_first</td></tr>  
## <tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td></tr>  
## <tr><td colspan="3" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">condWom</td><td>0.023</td><td>0.012</td></tr>  
## <tr><td style="text-align:left"></td><td>(0.059)</td><td>(0.037)</td></tr>  
## <tr><td style="text-align:left"></td><td></td><td></td></tr>  
## <tr><td style="text-align:left">Constant</td><td>1.394<sup>\*\*\*</sup></td><td>0.582<sup>\*\*\*</sup></td></tr>  
## <tr><td style="text-align:left"></td><td>(0.042)</td><td>(0.027)</td></tr>  
## <tr><td style="text-align:left"></td><td></td><td></td></tr>  
## <tr><td colspan="3" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">Observations</td><td>700</td><td>700</td></tr>  
## <tr><td style="text-align:left">R<sup>2</sup></td><td>0.0002</td><td>0.0002</td></tr>  
## <tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>-0.001</td><td>-0.001</td></tr>  
## <tr><td style="text-align:left">Residual Std. Error (df = 698)</td><td>0.780</td><td>0.493</td></tr>  
## <tr><td style="text-align:left">F Statistic (df = 1; 698)</td><td>0.146</td><td>0.105</td></tr>  
## <tr><td colspan="3" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"><em>Note:</em></td><td colspan="2" style="text-align:right"><sup>\*</sup>p<0.1; <sup>\*\*</sup>p<0.05; <sup>\*\*\*</sup>p<0.01</td></tr>  
## </table>

## Hyp 4: interaction between treatment and respondent’s status in underserved group

Similar to above, filters to those randomized to one of the barrier conditions

## women  
summary(lm(points\_rank\_rev ~ cond\*is\_woman,  
 data = barriers\_r))

##   
## Call:  
## lm(formula = points\_rank\_rev ~ cond \* is\_woman, data = barriers\_r)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.4591 -0.4162 0.5409 0.6169 0.6287   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.37126 0.06040 22.704 <2e-16 \*\*\*  
## condWom 0.08786 0.08648 1.016 0.310   
## is\_womanTRUE 0.04493 0.08467 0.531 0.596   
## condWom:is\_womanTRUE -0.12096 0.11846 -1.021 0.308   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7805 on 696 degrees of freedom  
## Multiple R-squared: 0.001821, Adjusted R-squared: -0.002481   
## F-statistic: 0.4233 on 3 and 696 DF, p-value: 0.7363

## minority  
summary(lm(points\_rank\_rev ~ relevel(factor(cond),  
 ref = "Wom")\*is\_minority,  
 data = barriers\_r))

##   
## Call:  
## lm(formula = points\_rank\_rev ~ relevel(factor(cond), ref = "Wom") \*   
## is\_minority, data = barriers\_r)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.4845 -0.3985 0.5254 0.6015 0.6420   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) 1.39847 0.04827  
## relevel(factor(cond), ref = "Wom")Min -0.04044 0.06952  
## is\_minorityTRUE 0.06618 0.09205  
## relevel(factor(cond), ref = "Wom")Min:is\_minorityTRUE 0.06033 0.13132  
## t value Pr(>|t|)   
## (Intercept) 28.972 <2e-16 \*\*\*  
## relevel(factor(cond), ref = "Wom")Min -0.582 0.561   
## is\_minorityTRUE 0.719 0.472   
## relevel(factor(cond), ref = "Wom")Min:is\_minorityTRUE 0.459 0.646   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7798 on 696 degrees of freedom  
## Multiple R-squared: 0.003561, Adjusted R-squared: -0.0007335   
## F-statistic: 0.8292 on 3 and 696 DF, p-value: 0.478

## minority (any black or Hispanic)  
summary(lm(points\_rank\_rev ~ relevel(factor(cond),  
 ref = "Wom")\*is\_minority\_sec,  
 data = barriers\_r))

##   
## Call:  
## lm(formula = points\_rank\_rev ~ relevel(factor(cond), ref = "Wom") \*   
## is\_minority\_sec, data = barriers\_r)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.5147 -0.4027 0.5224 0.5973 0.6360   
##   
## Coefficients:  
## Estimate  
## (Intercept) 1.40273  
## relevel(factor(cond), ref = "Wom")Min -0.03876  
## is\_minority\_secTRUE 0.07488  
## relevel(factor(cond), ref = "Wom")Min:is\_minority\_secTRUE 0.07585  
## Std. Error  
## (Intercept) 0.04555  
## relevel(factor(cond), ref = "Wom")Min 0.06565  
## is\_minority\_secTRUE 0.10559  
## relevel(factor(cond), ref = "Wom")Min:is\_minority\_secTRUE 0.14942  
## t value Pr(>|t|)  
## (Intercept) 30.794 <2e-16  
## relevel(factor(cond), ref = "Wom")Min -0.590 0.555  
## is\_minority\_secTRUE 0.709 0.478  
## relevel(factor(cond), ref = "Wom")Min:is\_minority\_secTRUE 0.508 0.612  
##   
## (Intercept) \*\*\*  
## relevel(factor(cond), ref = "Wom")Min   
## is\_minority\_secTRUE   
## relevel(factor(cond), ref = "Wom")Min:is\_minority\_secTRUE   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7797 on 696 degrees of freedom  
## Multiple R-squared: 0.003839, Adjusted R-squared: -0.0004548   
## F-statistic: 0.8941 on 3 and 696 DF, p-value: 0.4438

## maybe viz

## Hyp 5: interaction b/t other respondent demographics and treatment

Uses all respondents and pools the barriers treatment into single treatment

### Just world

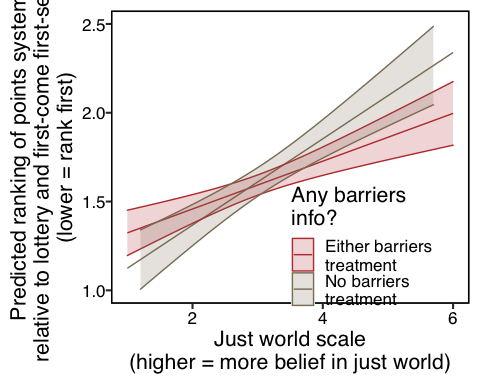
## first, just world  
summary(lm(points\_rank\_rev ~ is\_any\_barriers\*jw\_combined,  
 data = model\_df))

##   
## Call:  
## lm(formula = points\_rank\_rev ~ is\_any\_barriers \* jw\_combined,   
## data = model\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.8755 -0.4649 0.4013 0.5756 1.3391   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.11847 0.12925 16.391 < 2e-16 \*\*\*  
## is\_any\_barriersTRUE -0.30754 0.15860 -1.939 0.0528 .   
## jw\_combined -0.24293 0.03989 -6.090 1.58e-09 \*\*\*  
## is\_any\_barriersTRUE:jw\_combined 0.10837 0.04929 2.199 0.0281 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7705 on 1048 degrees of freedom  
## Multiple R-squared: 0.05341, Adjusted R-squared: 0.0507   
## F-statistic: 19.71 on 3 and 1048 DF, p-value: 1.96e-12

## binary regressing points as first  
## use lpm  
summary(lm(is\_points\_first ~ is\_any\_barriers\*jw\_combined,  
 data = model\_df))

##   
## Call:  
## lm(formula = is\_points\_first ~ is\_any\_barriers \* jw\_combined,   
## data = model\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.8909 -0.5220 0.2763 0.3968 0.8467   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.03847 0.08035 12.924 < 2e-16 \*\*\*  
## is\_any\_barriersTRUE -0.16464 0.09860 -1.670 0.0953 .   
## jw\_combined -0.14753 0.02480 -5.949 3.67e-09 \*\*\*  
## is\_any\_barriersTRUE:jw\_combined 0.05281 0.03064 1.723 0.0851 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.479 on 1048 degrees of freedom  
## Multiple R-squared: 0.05679, Adjusted R-squared: 0.05409   
## F-statistic: 21.03 on 3 and 1048 DF, p-value: 3.102e-13

### store ranking obj and do predict- predict ACTUAL RANK   
### and not reverse rank to avoid confusion  
points\_rank\_lm\_jw = lm(points\_rank ~ is\_any\_barriers\*jw\_combined,  
 data = model\_df)  
  
predict\_jw\_df = data.frame(is\_any\_barriers = rep(c(TRUE, FALSE),  
 each = 100),  
 jw\_combined = rep(seq(from = min(model\_df$jw\_combined),  
 to = max(model\_df$jw\_combined),  
 length.out = 100),   
 2))   
  
## predict across all values of jw  
predicted\_rank = predict(points\_rank\_lm\_jw,  
 newdata = predict\_jw\_df,  
 se.fit = TRUE)  
  
predicted\_rank\_4graph = cbind(predict\_jw\_df,  
 data.frame(estimate = predicted\_rank$fit,  
 se = predicted\_rank$se.fit)) %>%  
 mutate(lower = estimate -1.96 \*se,  
 upper = estimate + 1.96\*se,  
 barriers\_4graph = case\_when(is\_any\_barriers ~ "Either barriers\ntreatment",  
 TRUE ~ "No barriers\ntreatment"))  
  
jw\_prediction\_graph = ggplot(predicted\_rank\_4graph, aes(x = jw\_combined, y = estimate,   
 group = barriers\_4graph,  
 color = barriers\_4graph)) +  
 geom\_line() +  
 geom\_ribbon(aes(ymin = lower, ymax = upper, fill = barriers\_4graph),  
 alpha = 0.2) +  
 theme\_new() +  
 scale\_color\_manual(values = col\_map\_coarse) +  
 scale\_fill\_manual(values = col\_map\_coarse) +  
 theme(legend.position = c(0.7, 0.2),  
 legend.background = element\_blank()) +  
 xlab("Just world scale\n(higher = more belief in just world)") +  
 ylab("Predicted ranking of points system\nrelative to lottery and first-come first-served\n(lower = rank first)") +  
 ylim(1, 2.5) +  
 labs(fill = "Any barriers\ninfo?",  
 color = "Any barriers\ninfo?")  
  
jw\_prediction\_graph



ggsave("../output/jap\_paper/figs/jw\_prediction.png",  
 plot = jw\_prediction\_graph,  
 width = 12,  
 height = 8)

### Political affiliation

summary(lm(points\_rank\_rev ~ is\_any\_barriers\*relevel(factor(political\_affil\_buckets),  
 ref = "Democrat"),  
 data = model\_df))

##   
## Call:  
## lm(formula = points\_rank\_rev ~ is\_any\_barriers \* relevel(factor(political\_affil\_buckets),   
## ref = "Democrat"), data = model\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.6667 -0.5661 0.4294 0.4339 1.0676   
##   
## Coefficients:  
## Estimate  
## (Intercept) 1.57062  
## is\_any\_barriersTRUE -0.00453  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent -0.32387  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.09605  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican -0.63819  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.10747  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other -0.21533  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 0.13548  
## Std. Error  
## (Intercept) 0.05728  
## is\_any\_barriersTRUE 0.07035  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.10403  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.16576  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 0.10549  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.12671  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.20372  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 0.12995  
## t value  
## (Intercept) 27.421  
## is\_any\_barriersTRUE -0.064  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent -3.113  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.579  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican -6.050  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.848  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other -1.057  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 1.043  
## Pr(>|t|)  
## (Intercept) < 2e-16  
## is\_any\_barriersTRUE 0.9487  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.0019  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.5624  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 2.02e-09  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.3965  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.2908  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 0.2974  
##   
## (Intercept) \*\*\*  
## is\_any\_barriersTRUE   
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent \*\*   
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other   
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican \*\*\*  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent   
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other   
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.762 on 1044 degrees of freedom  
## Multiple R-squared: 0.07762, Adjusted R-squared: 0.07144   
## F-statistic: 12.55 on 7 and 1044 DF, p-value: 1.617e-15

## binary regressing points as first  
summary(lm(is\_points\_first ~ is\_any\_barriers\*relevel(factor(political\_affil\_buckets),  
 ref = "Democrat"),  
 data = model\_df))

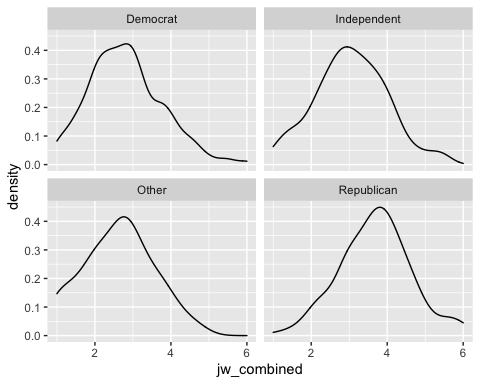
##   
## Call:  
## lm(formula = is\_points\_first ~ is\_any\_barriers \* relevel(factor(political\_affil\_buckets),   
## ref = "Democrat"), data = model\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.7500 -0.5195 0.3107 0.3190 0.6486   
##   
## Coefficients:  
## Estimate  
## (Intercept) 0.689266  
## is\_any\_barriersTRUE -0.008231  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent -0.169785  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.060734  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican -0.337914  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.034763  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other -0.082194  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 0.044204  
## Std. Error  
## (Intercept) 0.035944  
## is\_any\_barriersTRUE 0.044149  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.065283  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.104021  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 0.066199  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.079511  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.127841  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 0.081547  
## t value  
## (Intercept) 19.176  
## is\_any\_barriersTRUE -0.186  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent -2.601  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.584  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican -5.105  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.437  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other -0.643  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 0.542  
## Pr(>|t|)  
## (Intercept) < 2e-16  
## is\_any\_barriersTRUE 0.85214  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.00943  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.55944  
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 3.94e-07  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent 0.66205  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other 0.52040  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican 0.58789  
##   
## (Intercept) \*\*\*  
## is\_any\_barriersTRUE   
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent \*\*   
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Other   
## relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican \*\*\*  
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Independent   
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Other   
## is\_any\_barriersTRUE:relevel(factor(political\_affil\_buckets), ref = "Democrat")Republican   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4782 on 1044 degrees of freedom  
## Multiple R-squared: 0.06358, Adjusted R-squared: 0.0573   
## F-statistic: 10.13 on 7 and 1044 DF, p-value: 2.654e-12

### What’s relationship between political affiliation, just world scale, and pol ideology?

Republicans more evenly distributed throughout ranking distribution and have higher mean just world scale

Independents between dems and r on ideology scale

ggplot(model\_df, aes(x = jw\_combined)) +  
 geom\_density() +  
 facet\_wrap(~political\_affil\_buckets)



model\_df %>%  
 group\_by(political\_affil\_buckets) %>%  
 summarise(mean\_jw = mean(jw\_combined),  
 mean\_ideology\_higher\_conserv = mean(political\_ideology\_numeric,  
 na.rm = TRUE)) # remove na since coded prefer not/etc to na for numeric version

## # A tibble: 4 x 3  
## political\_affil\_buckets mean\_jw mean\_ideology\_higher\_conserv  
## \* <chr> <dbl> <dbl>  
## 1 Democrat 2.81 1.94  
## 2 Independent 3.05 2.81  
## 3 Other 2.62 2.55  
## 4 Republican 3.68 4.05

### Political ideology

Coded as numeric where 1 = very conservative, 5 = very liberal

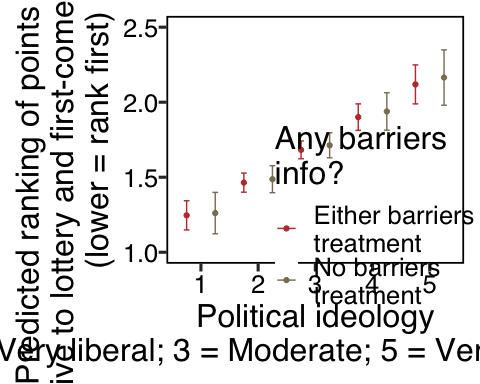
summary(lm(points\_rank\_rev ~ is\_any\_barriers\*political\_ideology\_numeric,  
 data = model\_df))

##   
## Call:  
## lm(formula = points\_rank\_rev ~ is\_any\_barriers \* political\_ideology\_numeric,   
## data = model\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.7540 -0.5358 0.2611 0.4869 1.1645   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) 1.964799 0.101733  
## is\_any\_barriersTRUE 0.007342 0.124351  
## political\_ideology\_numeric -0.225851 0.035681  
## is\_any\_barriersTRUE:political\_ideology\_numeric 0.007665 0.043599  
## t value Pr(>|t|)   
## (Intercept) 19.313 < 2e-16 \*\*\*  
## is\_any\_barriersTRUE 0.059 0.953   
## political\_ideology\_numeric -6.330 3.66e-10 \*\*\*  
## is\_any\_barriersTRUE:political\_ideology\_numeric 0.176 0.860   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.752 on 1031 degrees of freedom  
## (17 observations deleted due to missingness)  
## Multiple R-squared: 0.1013, Adjusted R-squared: 0.09869   
## F-statistic: 38.74 on 3 and 1031 DF, p-value: < 2.2e-16

## binary regressing points as first  
summary(lm(is\_points\_first ~ is\_any\_barriers\*political\_ideology\_numeric,  
 data = model\_df))

##   
## Call:  
## lm(formula = is\_points\_first ~ is\_any\_barriers \* political\_ideology\_numeric,   
## data = model\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.7903 -0.5380 0.2117 0.3366 0.7143   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) 0.916459 0.063881  
## is\_any\_barriersTRUE -0.003209 0.078084  
## political\_ideology\_numeric -0.126156 0.022405  
## is\_any\_barriersTRUE:political\_ideology\_numeric 0.001210 0.027377  
## t value Pr(>|t|)   
## (Intercept) 14.346 < 2e-16 \*\*\*  
## is\_any\_barriersTRUE -0.041 0.967   
## political\_ideology\_numeric -5.631 2.31e-08 \*\*\*  
## is\_any\_barriersTRUE:political\_ideology\_numeric 0.044 0.965   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4722 on 1031 degrees of freedom  
## (17 observations deleted due to missingness)  
## Multiple R-squared: 0.08419, Adjusted R-squared: 0.08152   
## F-statistic: 31.59 on 3 and 1031 DF, p-value: < 2.2e-16

## generate pred using real rank  
points\_rank\_lm\_polid = lm(points\_rank ~ is\_any\_barriers\*political\_ideology\_numeric,  
 data = model\_df)  
  
predict\_polid\_df = data.frame(is\_any\_barriers = rep(c(TRUE, FALSE),  
 each = 5),  
 political\_ideology\_numeric = rep(seq(from = 1,  
 to = 5),   
 2))   
  
## predict across all values of jw  
predicted\_rank\_pol = predict(points\_rank\_lm\_polid,  
 newdata = predict\_polid\_df,  
 se.fit = TRUE)  
  
predicted\_rank\_4graph\_pol = cbind(predict\_polid\_df,  
 data.frame(estimate = predicted\_rank\_pol$fit,  
 se = predicted\_rank\_pol$se.fit)) %>%  
 mutate(lower = estimate -1.96 \*se,  
 upper = estimate + 1.96\*se,  
 barriers\_4graph = case\_when(is\_any\_barriers ~ "Either barriers\ntreatment",  
 TRUE ~ "No barriers\ntreatment"))  
  
pol\_prediction\_graph = ggplot(predicted\_rank\_4graph\_pol, aes(x = factor(political\_ideology\_numeric), y = estimate,   
 group = barriers\_4graph,  
 color = barriers\_4graph)) +  
 geom\_point(position = position\_dodge(width = 1)) +  
 geom\_errorbar(aes(ymin = lower, ymax = upper, color = barriers\_4graph),  
 width = 0.2,  
 position = position\_dodge(width = 1)) +  
 theme\_new(base\_size = 24) +  
 scale\_color\_manual(values = col\_map\_coarse) +  
 theme(legend.position = c(0.7, 0.2),  
 legend.background = element\_blank()) +  
 xlab("Political ideology\n(1 = Very liberal; 3 = Moderate; 5 = Very conservative)") +  
 ylab("Predicted ranking of points system\nrelative to lottery and first-come first-served\n(lower = rank first)") +  
 ylim(1, 2.5) +  
 labs(color = "Any barriers\ninfo?")  
  
pol\_prediction\_graph

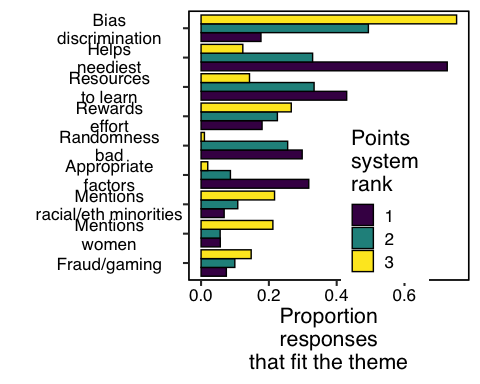


ggsave("../output/jap\_paper/figs/pol\_prediction.png",  
 plot = pol\_prediction\_graph,  
 width = 12,  
 height = 8)   
  
  
## could relabel with actual values

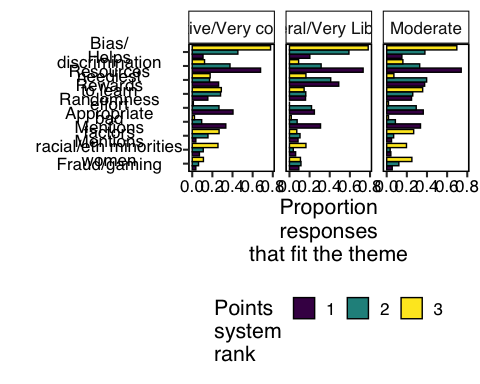
## Merge free responses

### Table summarizing coding

## first, left join fr onto: (1) analytic sample (so not looking at proportions for those excluded  
## and (2) limited to treatment and demographics  
## joining by prol\_id rather than prol\_id\_clean since  
## we didn't do id cleaning before writing for coding  
colnames(fr\_coded\_both) = gsub("\\.+", "\_", colnames(fr\_coded\_both))  
  
  
fr\_thematic\_cols = setdiff(colnames(fr\_coded\_both),  
 c("prol\_id", "choice",  
 "fr\_why\_first\_over\_second",  
 "fr\_why\_second\_over\_third",  
 "coder",  
 "Nonsensical\_1\_yes\_0\_no\_",  
 "Other\_reasons",  
 "Comments\_on\_other\_reasons"))  
  
model\_df\_wresp= model\_df %>%  
 select(prol\_id, prol\_id\_clean,   
 cond, points\_rank,  
 is\_woman, raceeth,   
 is\_minority,   
 political\_affil,  
 political\_affil\_buckets,  
 political\_ideology,  
 jw\_combined) %>%  
 left\_join(fr\_coded\_both, by = "prol\_id") %>%  
 mutate(is\_matched\_tofr = ifelse(prol\_id %in% fr\_coded\_both$prol\_id,  
 TRUE, FALSE)) %>%  
 filter(is\_matched\_tofr) # one not matched due to not filling out fr but having non-missing DV  
   
  
## there are 9ish that have NA due to being initially coded as nonsensical   
## but that heather reversed codes for--- judgment was that they didnt fit theme so code  
## their thematic cols to 0  
model\_df\_wresp[, fr\_thematic\_cols][is.na(model\_df\_wresp[, fr\_thematic\_cols])] <- 0  
  
## find overall proportions  
prop\_themes\_all = data.frame(proportion\_resp = colMeans(model\_df\_wresp[, fr\_thematic\_cols])) %>%  
 arrange(desc(proportion\_resp))  
prop\_themes\_all$theme = rownames(prop\_themes\_all)  
write.table(prop\_themes\_all,   
 file = "../output/jap\_paper/tables/overall\_fr\_themes.txt",  
 sep = ",", quote = FALSE, row.names = F)  
write.csv(prop\_themes\_all,   
 file = "../output/jap\_paper/tables/overall\_fr\_themes.csv",  
 quote = FALSE, row.names = F)  
  
## proportions by condition  
prop\_by\_cond = model\_df\_wresp %>%  
 group\_by(cond) %>%  
 summarise\_at(.vars = fr\_thematic\_cols, mean) # for this, among no\_info, since half randomzied to points system  
 # about women or minorities, those mention that   
write.table(prop\_by\_cond,   
 file = "../output/jap\_paper/tables/themes\_by\_treatcondition.txt",  
 sep = ",", quote = FALSE, row.names = F)  
write.csv(prop\_by\_cond,   
 file = "../output/jap\_paper/tables/themes\_by\_treatcondition.csv",  
 quote = FALSE, row.names = F)  
  
## proportions by rank for points system  
prop\_by\_rank = model\_df\_wresp %>%  
 group\_by(points\_rank) %>%  
 summarise\_at(.vars = fr\_thematic\_cols, mean) # for this, among no\_info, since half randomzied to points system  
 # about women or minorities, those mention that   
write.table(prop\_by\_rank,   
 file = "../output/jap\_paper/tables/themes\_by\_pointsranking.txt",  
 sep = ",", quote = FALSE, row.names = F)  
write.csv(prop\_by\_rank,   
 file = "../output/jap\_paper/tables/themes\_by\_pointsranking.csv",  
 quote = FALSE, row.names = F)  
  
  
  
## visualize using shorthand  
prop\_by\_rank\_long = reshape2::melt(prop\_by\_rank, id.var = "points\_rank") %>%  
 mutate(theme\_short =   
 case\_when(grepl("^Propensity", variable) ~ "Fraud/gaming",  
 grepl("^Ability", variable) ~ "Helps\nneediest",  
 grepl("personal\_resources", variable) ~ "Resources\nto learn",  
 grepl("bias\_discrimination", variable) ~ "Bias\ndiscrimination",  
 grepl("reward\_applicant", variable) ~ "Rewards\neffort",  
 grepl("randomness", variable) ~ "Randomness\nbad",  
 grepl("appropriate", variable) ~ "Appropriate\nfactors",  
 grepl("^Women", variable) ~ "Mentions\nwomen",  
 grepl("^Racial", variable) ~ "Mentions\nracial/eth minorities"))  
  
  
ggplot(prop\_by\_rank\_long, aes(x = reorder(factor(theme\_short), value), y = value,  
 group = factor(points\_rank),  
 fill = factor(points\_rank))) +  
 geom\_bar(stat = "identity", position = "dodge",  
 color = "black") +  
 coord\_flip() +  
 theme\_new() +  
 scale\_fill\_viridis(discrete = TRUE) +  
 ylab("Proportion\nresponses\nthat fit the theme") +  
 xlab("") +  
 theme(legend.position = c(0.7, 0.3)) +  
 labs(fill = "Points\nsystem\nrank")



ggsave("../output/jap\_paper/figs/prop\_frthemes\_byrank.png",  
 plot = last\_plot(),  
 width = 12,  
 height = 8)   
  
## breakdown by political ideology  
prop\_by\_rank\_byideol = reshape2::melt(model\_df\_wresp %>%  
 mutate(ideology\_broad = case\_when(  
 grepl("Conservative", political\_ideology) ~   
 "Conservative/Very conservative",  
 grepl("Liberal", political\_ideology) ~ "Liberal/Very Liberal",  
 grepl("Moderate", political\_ideology) ~ "Moderate",  
 TRUE ~ "Other"  
 )) %>%  
 filter(ideology\_broad != "Other") %>%  
 group\_by(ideology\_broad, points\_rank) %>%  
 summarise\_at(.vars = fr\_thematic\_cols, mean),  
 id.vars = c("ideology\_broad", "points\_rank")) %>%  
 mutate(theme\_short =   
 case\_when(grepl("^Propensity", variable) ~ "Fraud/gaming",  
 grepl("^Ability", variable) ~ "Helps\nneediest",  
 grepl("personal\_resources", variable) ~ "Resources\nto learn",  
 grepl("bias\_discrimination", variable) ~ "Bias/\ndiscrimination",  
 grepl("reward\_applicant", variable) ~ "Rewards\neffort",  
 grepl("randomness", variable) ~ "Randomness\nbad",  
 grepl("appropriate", variable) ~ "Appropriate\nfactors",  
 grepl("^Women", variable) ~ "Mentions\nwomen",  
 grepl("^Racial", variable) ~ "Mentions\nracial/eth minorities"))  
  
ggplot(prop\_by\_rank\_byideol, aes(x = reorder(factor(theme\_short), value), y = value,  
 group = factor(points\_rank),  
 fill = factor(points\_rank))) +  
 geom\_bar(stat = "identity", position = "dodge",  
 color = "black") +  
 coord\_flip() +  
 theme\_new() +  
 scale\_fill\_viridis(discrete = TRUE) +  
 ylab("Proportion\nresponses\nthat fit the theme") +  
 xlab("") +  
 theme(legend.position = "bottom") +  
 labs(fill = "Points\nsystem\nrank") +  
 facet\_wrap(~ideology\_broad, ncol = 3)



ggsave("../output/jap\_paper/figs/prop\_frthemes\_byrank\_byideology.png",  
 plot = last\_plot(),  
 width = 12,  
 height = 8)

### Sheet to upload

write.csv(model\_df\_wresp %>%  
 arrange(desc(points\_rank),  
 political\_affil\_buckets),  
 "../output/fr\_forquotes.csv",  
 row.names = FALSE)

### Bivariate corr

## add later; since some are not continuous etc