Final analysis code

Heather Kappes, Crystal Hall, Rebecca Johnson, Simone Zhang

01/06/2020

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)))
```

```r
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"
```

```r
## 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)))

# Inspect the unmerged observations with no match in the prolific data
# Results in 22 observations that don't merge
table(raw_data_wp$is_failmerge_prol, useNA = "always")
```

```
##
## FALSE  TRUE  <NA>
##  1225    22     0
```

```r
raw_data_wp_unmatched = raw_data_wp %>%
  filter(is_failmerge_prol)

table(raw_data_wp_unmatched$is_prolid_wronglength, useNA = "always")
```

```
##
```

```
## FALSE  TRUE  <NA>
##     4     9     9
```

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 t
                                      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(...)`.
```

```
# Review flags
table(raw_data_wp_intermed$political_affil, useNA = "always")
```

```
##
##          Democrat       Independent     No preference       Other party
##               611               282                51                23
## Prefer not to say        Republican              <NA>
##                 4               257                19
```

```
table(raw_data_wp_cleanvar$political_affil_buckets, useNA = "always")
```

```
##
##    Democrat Independent       Other  Republican        <NA>
##         611         282          97         257           0
```

```
temp <- raw_data_wp_cleanvar %>%
  select(contains("jw"))
```

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

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 r
##
##                            The researcher definitely wanted me to say first-come first-served was the r
##
##                                                The researcher definitely wanted me to say lottery was the r
##
##                                                 The researcher definitely wanted me to say points was the r
##
##                            The researcher might have wanted me to say first-come first-served was the r
##
##                                                The researcher might have wanted me to say lottery was the r
##
##                                                 The researcher might have wanted me to say points was the r
##
## The researcher probably wanted me to say one method was the most fair, but I don't know which method
##
##
## $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 r
```

```
##
##                          The researcher definitely wanted me to say first-come first-served was the r
##
##                                              The researcher definitely wanted me to say lottery was the r
##
##                                               The researcher definitely wanted me to say points was the r
##
##                          The researcher might have wanted me to say first-come first-served was the r
##
##                                              The researcher might have wanted me to say lottery was the r
##
##                                               The researcher might have wanted me to say points was the r
##
## The researcher probably wanted me to say one method was the most fair, but I don't know which method
##
##
## $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 r
##                                                                                                     (
##                           The researcher definitely wanted me to say first-come first-served was the r
##                                                                                                     (
##                                      The researcher definitely wanted me to say lottery was the r
##                                                                                                     (
```

```
##                                                 The researcher definitely wanted me to say points was the r
##                                                                                                            (
##                                The researcher might have wanted me to say first-come first-served was the r
##                                                                                                            (
##                                               The researcher might have wanted me to say lottery was the r
##                                                                                                            (
##                                                The researcher might have wanted me to say points was the r
##                                                                                                            (
## The researcher probably wanted me to say one method was the most fair, but I don't know which method
##                                                                                                            (
##
## $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
```

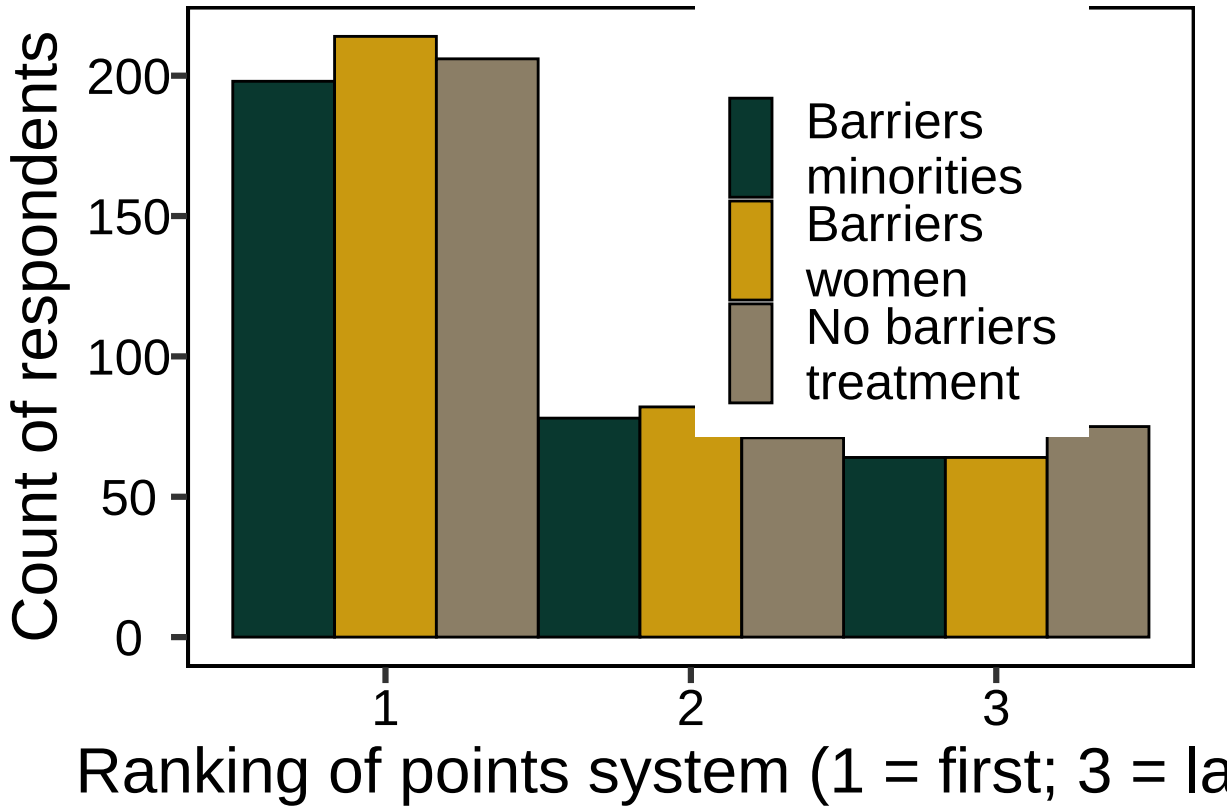
```
table(model_df$is_any_barriers)
```

```
##
## 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) +
```

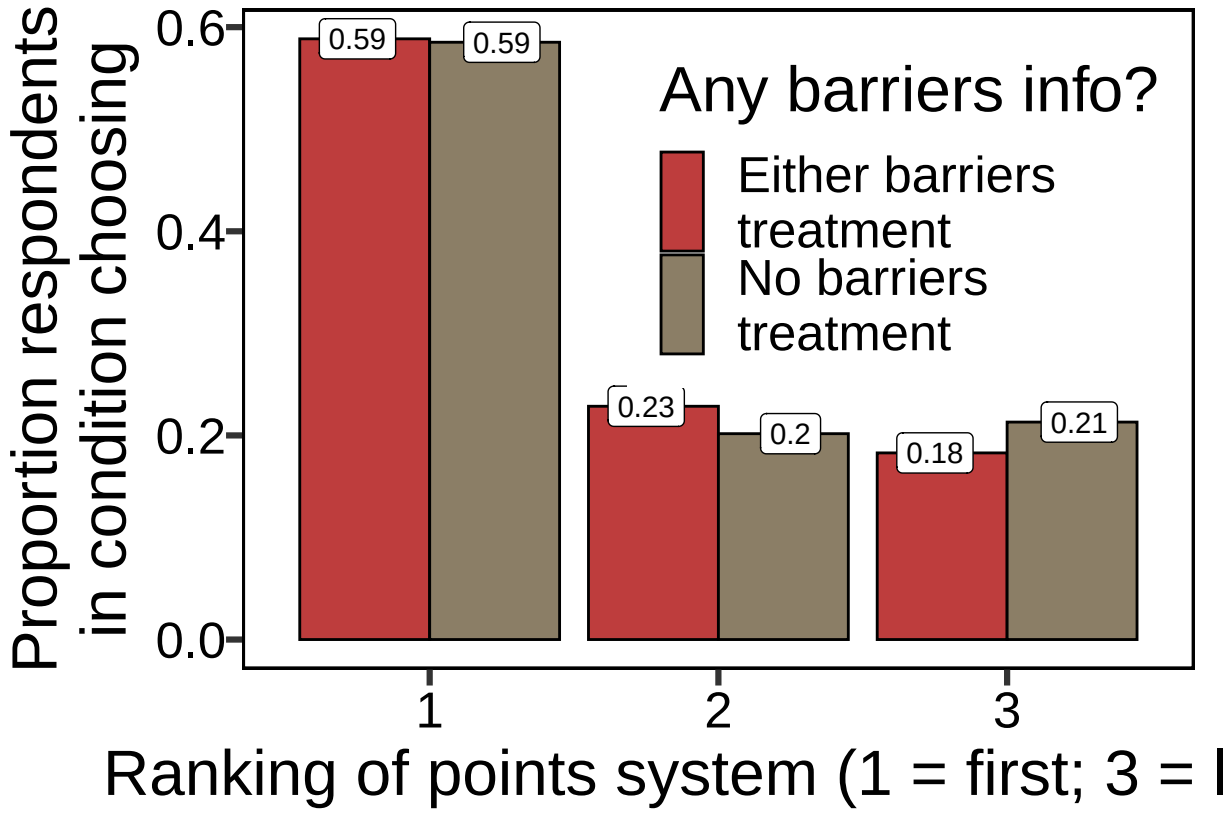
```
  xlab("Ranking of points system (1 = first; 3 = last)") +
  theme(legend.position = c(0.7, 0.7)) +
  labs(fill = "") +
  ylab("Count of respondents") +
  scale_fill_manual(values = col_map_detailed)

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
```



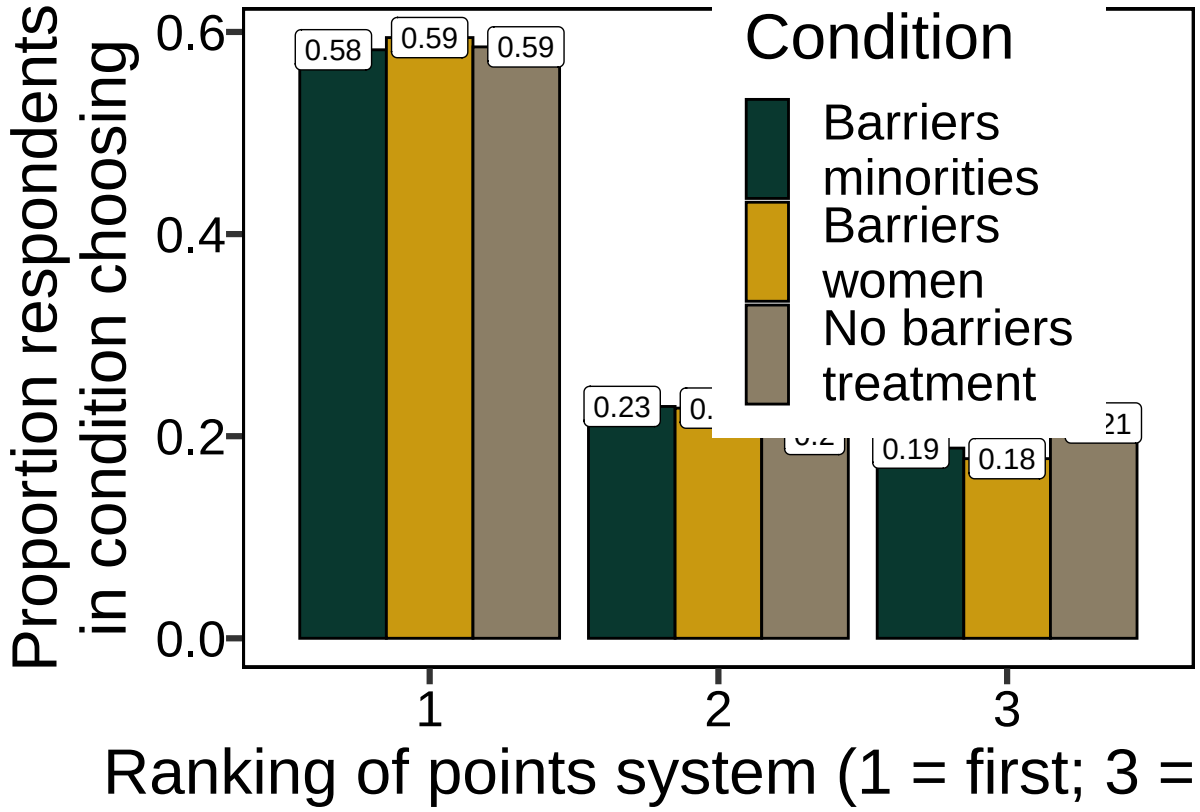

```
ggsave("../output/jap_paper/figs/rankings_prop_2groups.png",
       plot = prop_choose_plot,
       width = 12,
       height = 8)


## 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
```

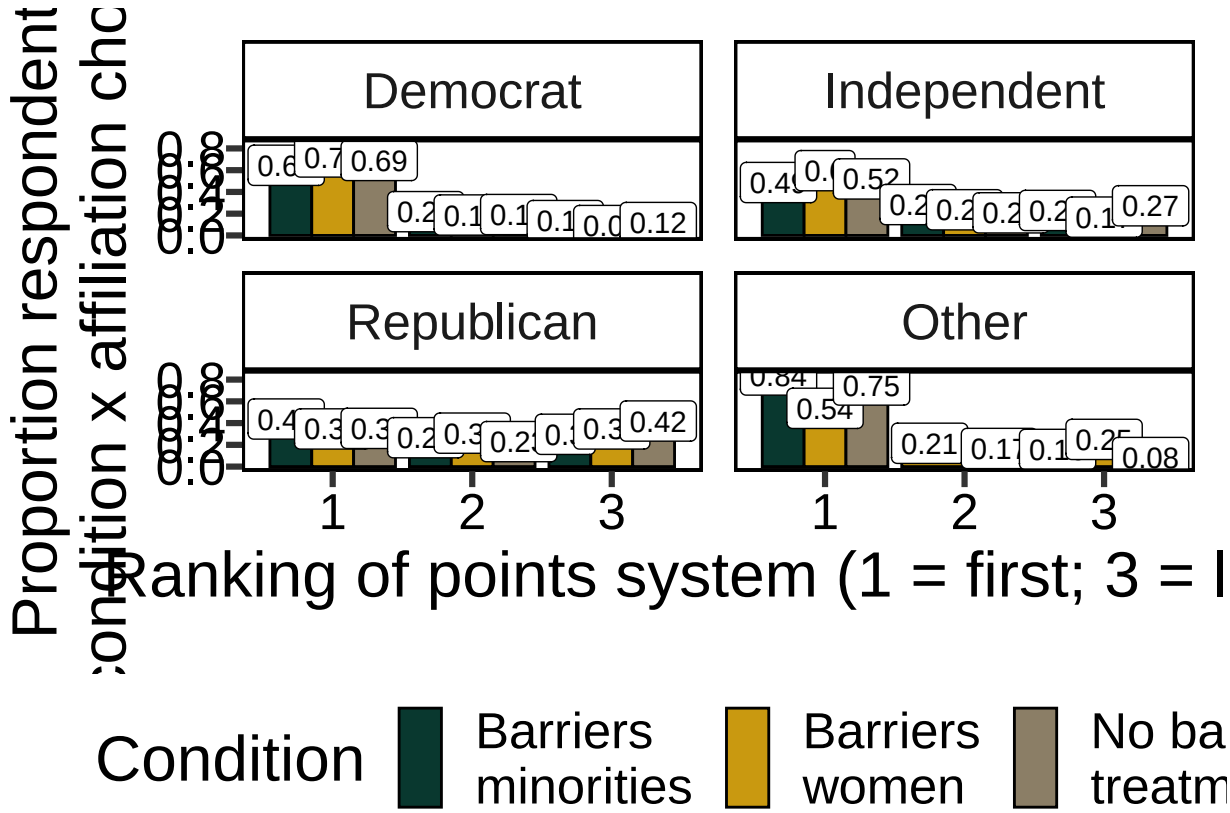



```
ggsave("../output/jap_paper/figs/rankings_prop_3groups.png",
       plot = prop_choose_plot_detail,
       width = 12,
       height = 8)

## 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
## `summarise()` has grouped output by 'cond_4graph'. You can override using the `.groups` argument.
## Joining, by = c("cond_4graph", "political_affil_buckets")
```

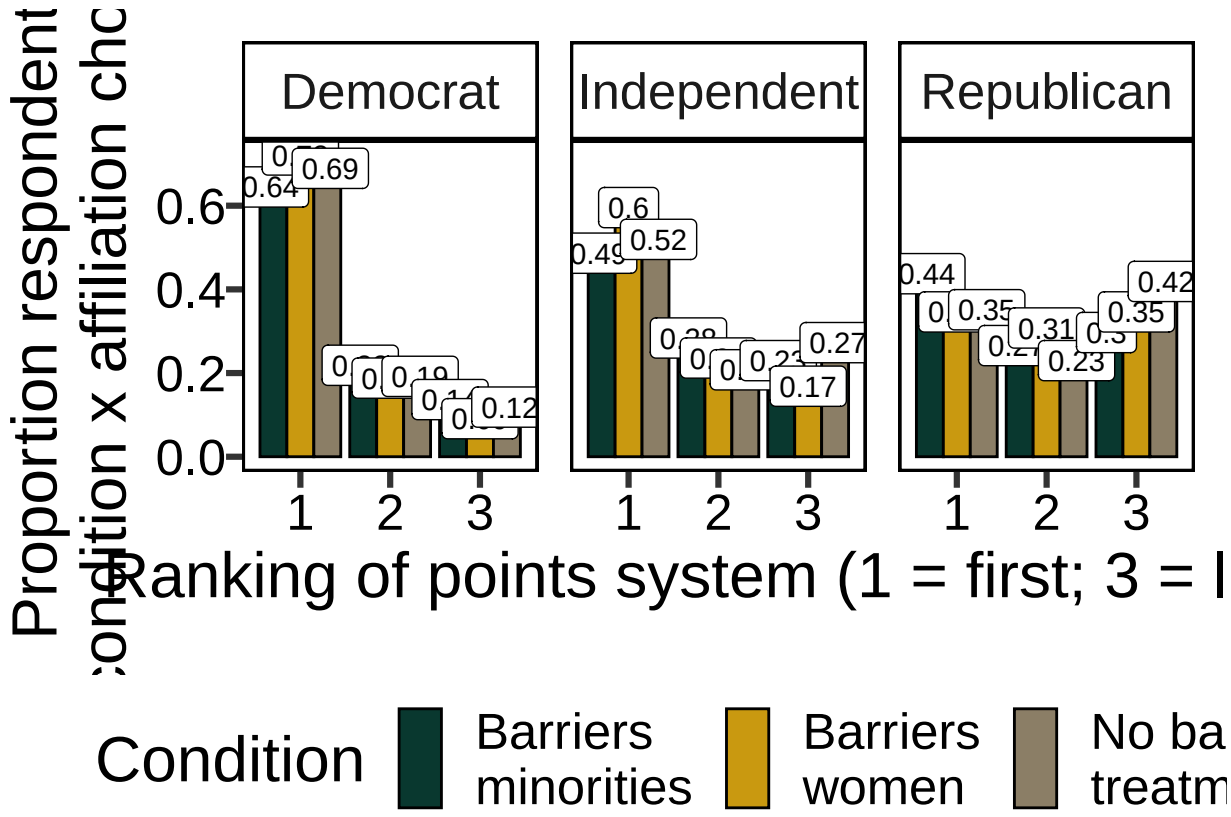
```
prop_choose_plot_detail_waffil <- ggplot(prop_choose_detail_waffil, aes(x = points_rank,
                        y = prop_choose,
                        group = cond_4graph,
                        fill = cond_4graph)) +
  geom_bar(stat = "identity",
           position = "dodge", col = "black") +
  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
```



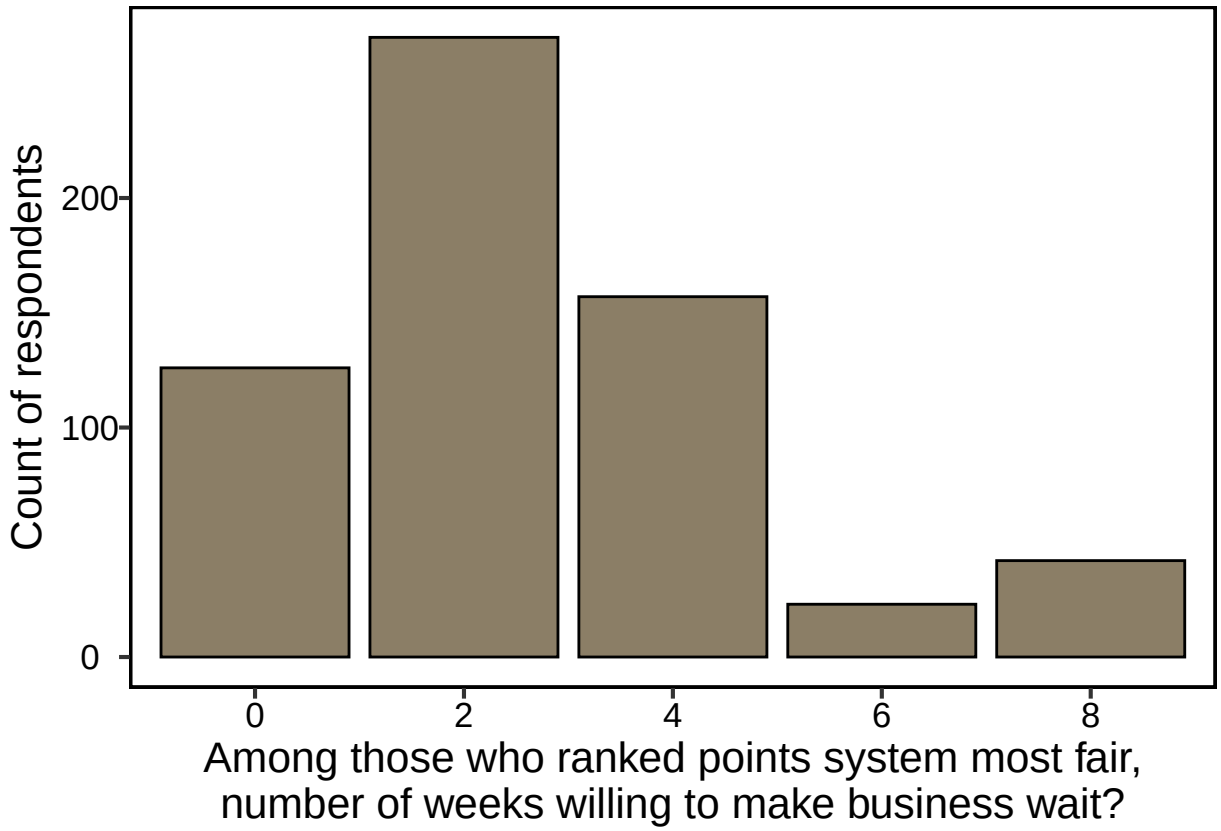

```
ggsave("../output/jap_paper/figs/rankings_prop_3groups_wpoliticalaffil.png",
       plot = prop_choose_plot_detail_waffil,
       width = 12,
       height = 8)

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></t
## <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"
## <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><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">
## <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">
## </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/minor
# that underserved group got no advantage) to 0 (not sure) to 3 (very importantthat underserved group g
```

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></t
## <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"
## <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><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"
## <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"
## </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

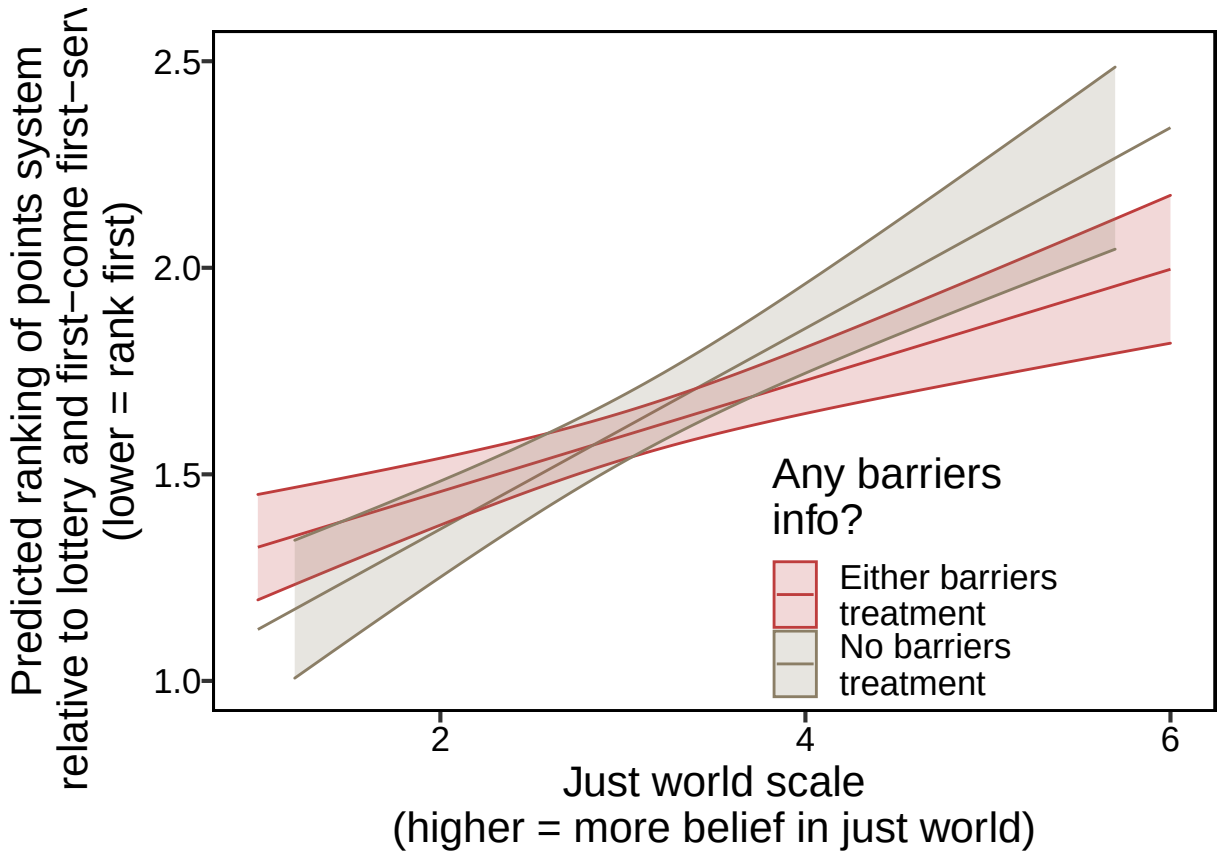
```
## first, just world
summary(lm(points_rank_rev ~ is_any_barriers*jw_combined,
           data = model_df))
```

```
##
## 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
```

```r
### 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 = ra
  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
```

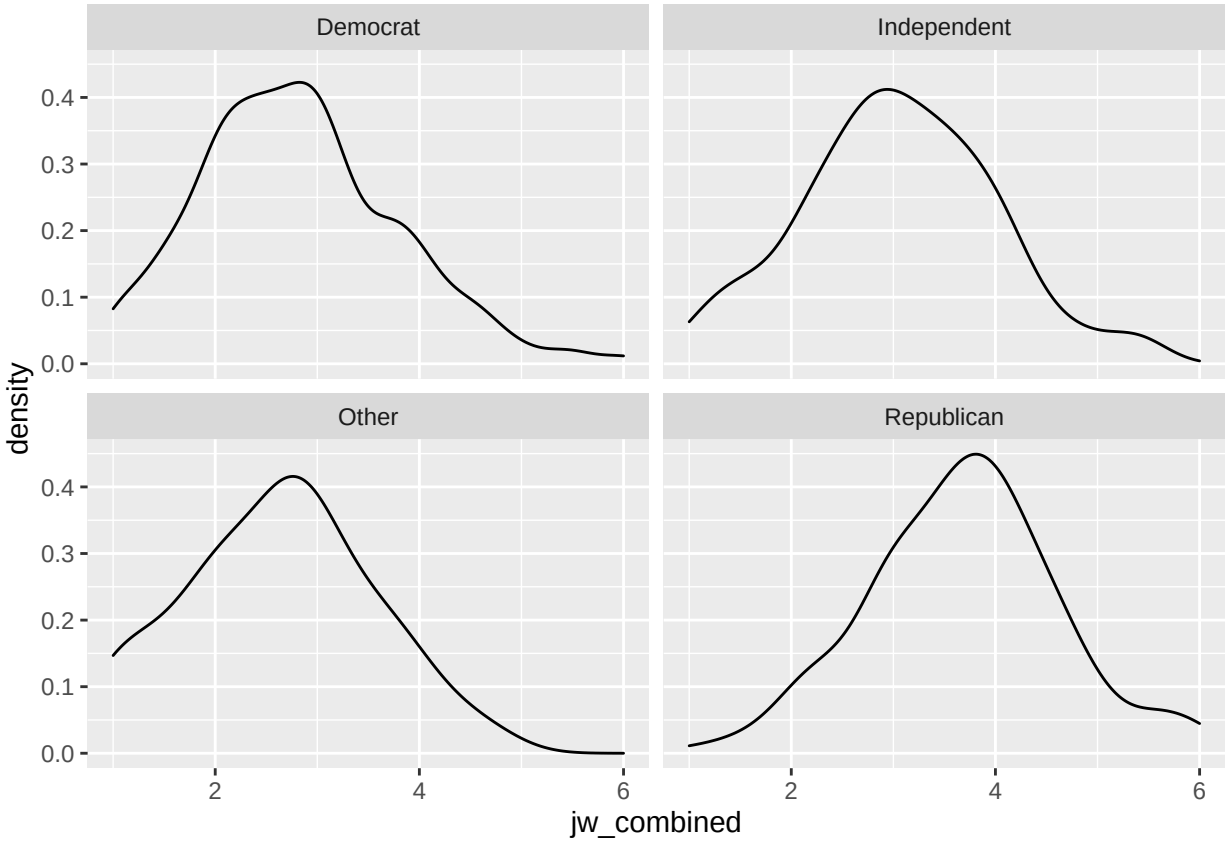
```
##
## Residual standard error: 0.4782 on 1044 degrees of freedom
## Multiple R-squared:  0.06358,    Adjusted R-squared:  0.0573
## F-statistic: 10.13 on 7 and 1044 DF,  p-value: 2.654e-12
```

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 t
```

```
## # 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))
```

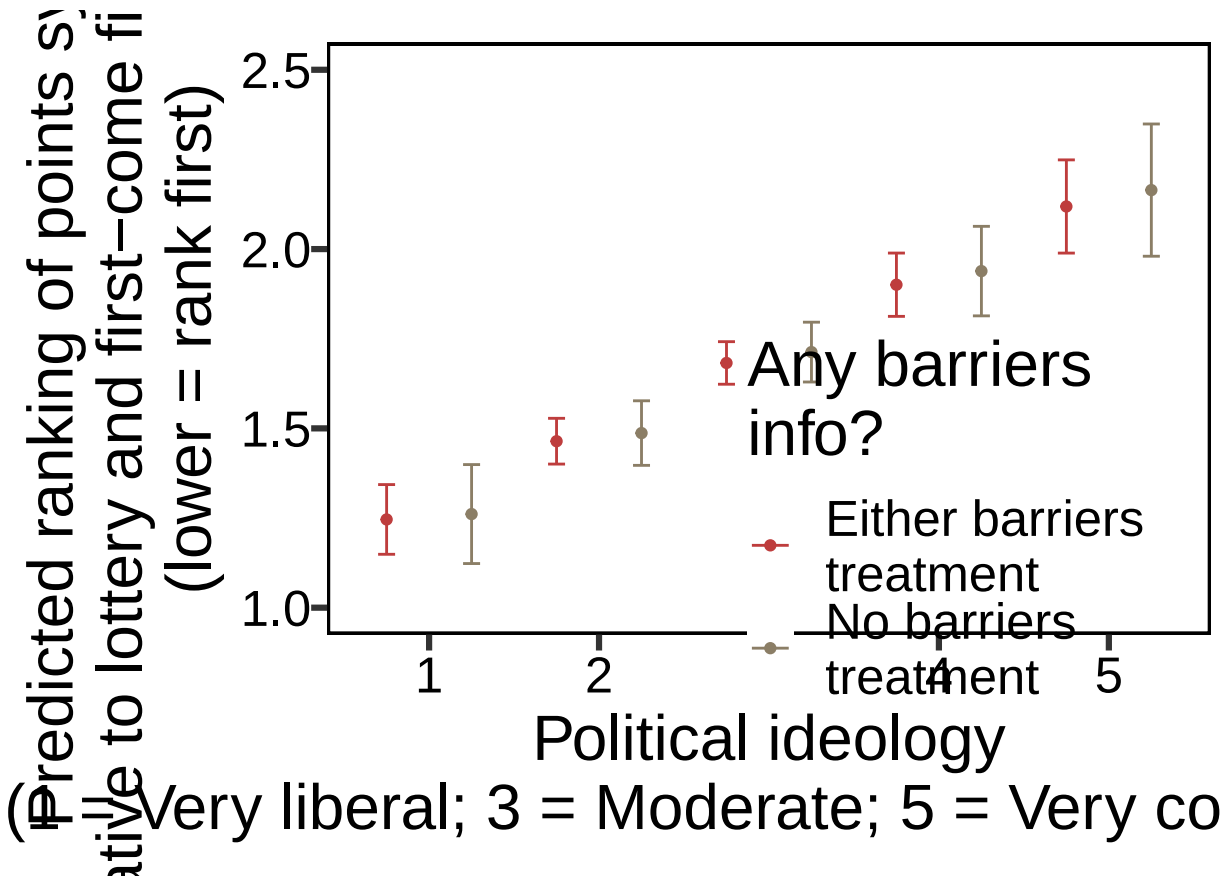
```
##
## Call:
## lm(formula = points_rank_rev ~ is_any_barriers * political_ideology_numeric,
##     data = model_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7540 -0.5358  0.2611  0.4869  1.1645
##
## Coefficients:
##                                             Estimate Std. 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
```

```r
## 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 =
                                  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 = ra
  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
```

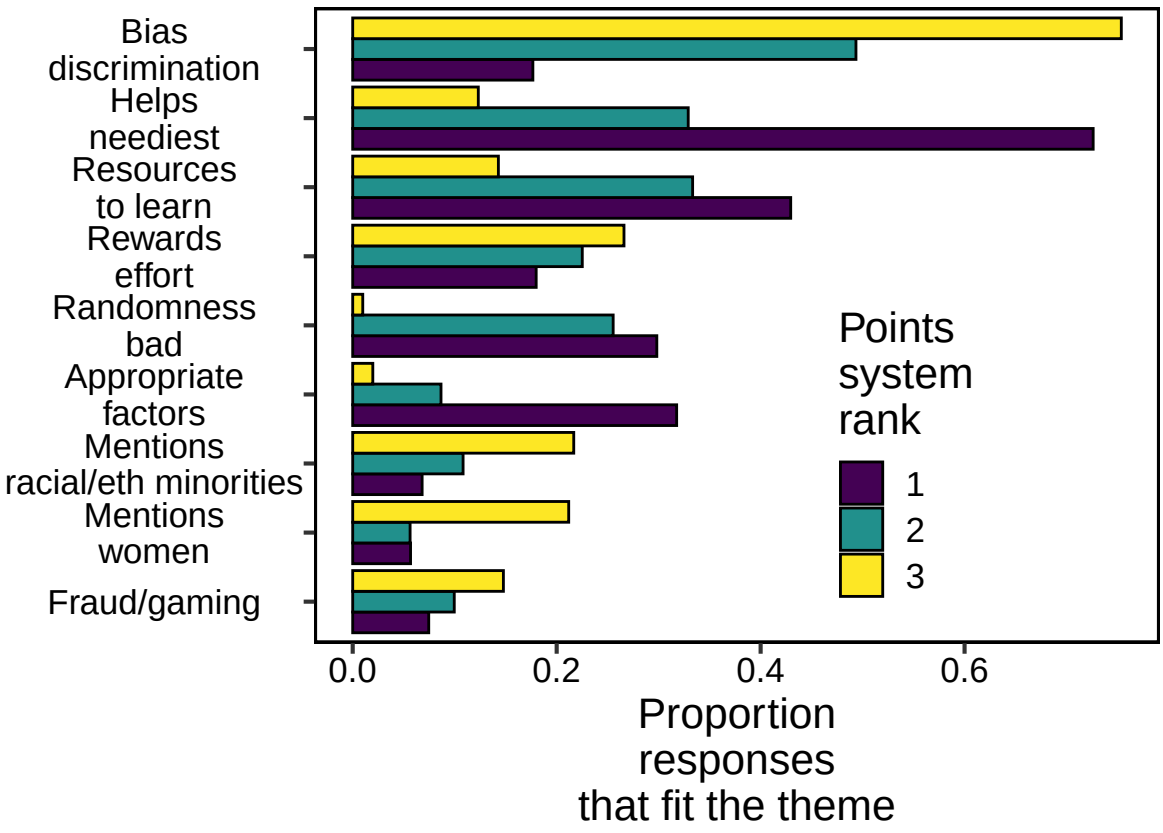
Merge free responses

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_",
```

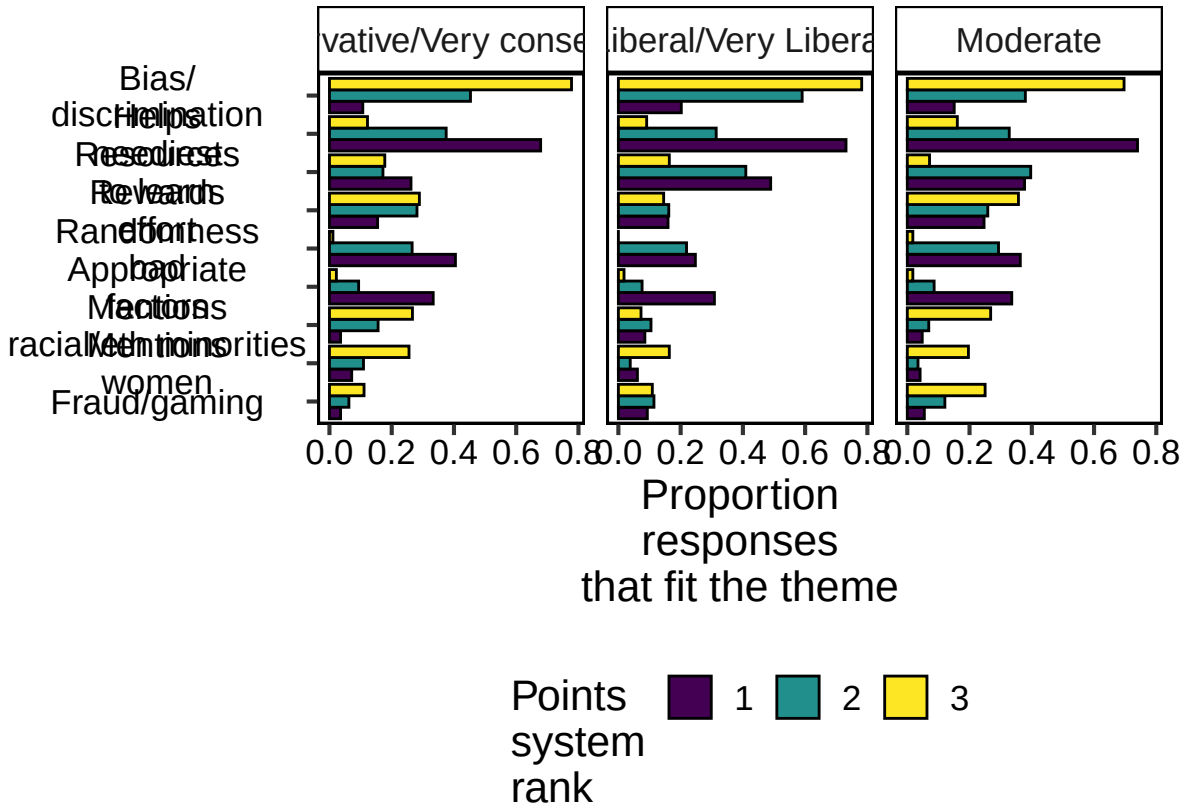
```
                            "Other_reasons",
                            "Comments_on_other_reasons"))

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


## 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 randomzi
            # 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 randomzi
            # 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
```