chatbot_paper_jop.R

moffer

2024-11-29

```
### R code from vignette source 'chatbot_paper_jop.Rnw'

###################################################
### code chunk number 1: setup
###################################################
#| eval = TRUE,
#| echo = FALSE,
#| results = 'hide',
#| message = FALSE

require(knitr, quietly = TRUE)

opts_chunk$set(cache = FALSE,
               cache.path = 'cache_paper/',
               fig.path = 'figures_paper/',
               tidy = TRUE,
               echo = FALSE,
               warning = FALSE,
               message = FALSE,
               fig.pos = '!ht',
               dev = 'pdf',
               dpi=200)

options(width = 110, knitr.kable.NA = '')

options(modelsummary_factory_latex = 'kableExtra')

knit_hooks$set(inline = function(x) {
  prettyNum(x, big.mark=',')
})

# load colors
cbPalette <- c('#999999', '#E69F00', '#56B4E9', '#009E73', '#F0E442', '#0072B2', '#D55E00', '#CC79A7')



###################################################
### code chunk number 2: load
###################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| results = 'hide'
```

```r
# load functions
source('utils.R')
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows

## Warning: package 'modelsummary' was built under R version 4.3.3

## `modelsummary` 2.0.0 now uses `tinytable` as its default table-drawing backend. Learn more at:
##   https://vincentarelbundock.github.io/tinytable/
##
## Revert to `kableExtra` for one session:
##
##   options(modelsummary_factory_default = 'kableExtra')
##   options(modelsummary_factory_latex = 'kableExtra')
##   options(modelsummary_factory_html = 'kableExtra')
##
## Silence this message forever:
##
##   config_modelsummary(startup_message = FALSE)

## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```r
### read in evaluation data
df_eval <- readRDS('clean_evaluation_data.rds')
eval_n <- nrow(df_eval)

### read in concerns data
df_concerns <- readRDS('clean_learning_concerns_data.rds')
concerns_n <- nrow(df_concerns)
learn_n <- length(unique(df_concerns$user_id))

df_nigeria <- df_eval[df_eval$country1 == 'nigeria', ]
df_kenya <- df_eval[df_eval$country1 == 'kenya', ]

# sample size in each country
nigeria_n <- nrow(df_nigeria)
kenya_n <- nrow(df_kenya)

# load colors
cbPalette <- c('#999999', '#E69F00', '#56B4E9', '#009E73', '#F0E442', '#0072B2',
               '#D55E00', '#CC79A7')
```

```r
## ggplot theme
vcf_theme <- function(){
  theme(panel.background = element_rect(fill = 'white',
                                        colour = NA),
        panel.border = element_rect(fill = NA,
                                    colour = 'grey20'),
        panel.grid = element_line(colour = 'grey92'),
        panel.grid.minor = element_line(size = rel(0.5)),
        strip.background = element_rect(fill = 'grey85',
                                        colour = 'grey20'),
        legend.title = element_text(size = 14),#element_blank(),
        legend.text = element_text(size = 12),
        plot.title = element_text(size = 14),
        legend.position = 'bottom',
        axis.text = element_text(size=12),
        axis.title = element_text(size = 14),
        strip.text = element_text(size = 12),
        plot.caption = element_text(size = 12)
  )
}


# just pretest willingness_2s
covariate_list_short <- c('willingness_f', 'get_vaccinated_f')



#####################################################
### code chunk number 3: main_alt
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| fig.align='center',
#| fig.height=6,
#| fig.width = 13,
#| prefix.string = 'fig',
#| fig.cap=
#| paste0('\\textbf{Average treatment effects of PSA and concern-addressing
#|  chatbot interventions compared to control.}',
#| ' The sample is users in the evaluation stage, $n = $',
#| prettyNum(eval_n, big.mark = ','),
#|  '. Estimates are produced from a linear estimator, controlling for pre-test
#|  response;
#|  error bars represent symmetric 95\\% confidence intervals with robust
#|  standard errors.'),
#| strip.white=TRUE,
#| results='asis'

covariate_lin <- as.formula(paste('~' , paste0(covariate_list_short,
                                               collapse = ' + ')))

# Comparison to PSA
df_eval_alt <- df_eval |>
```

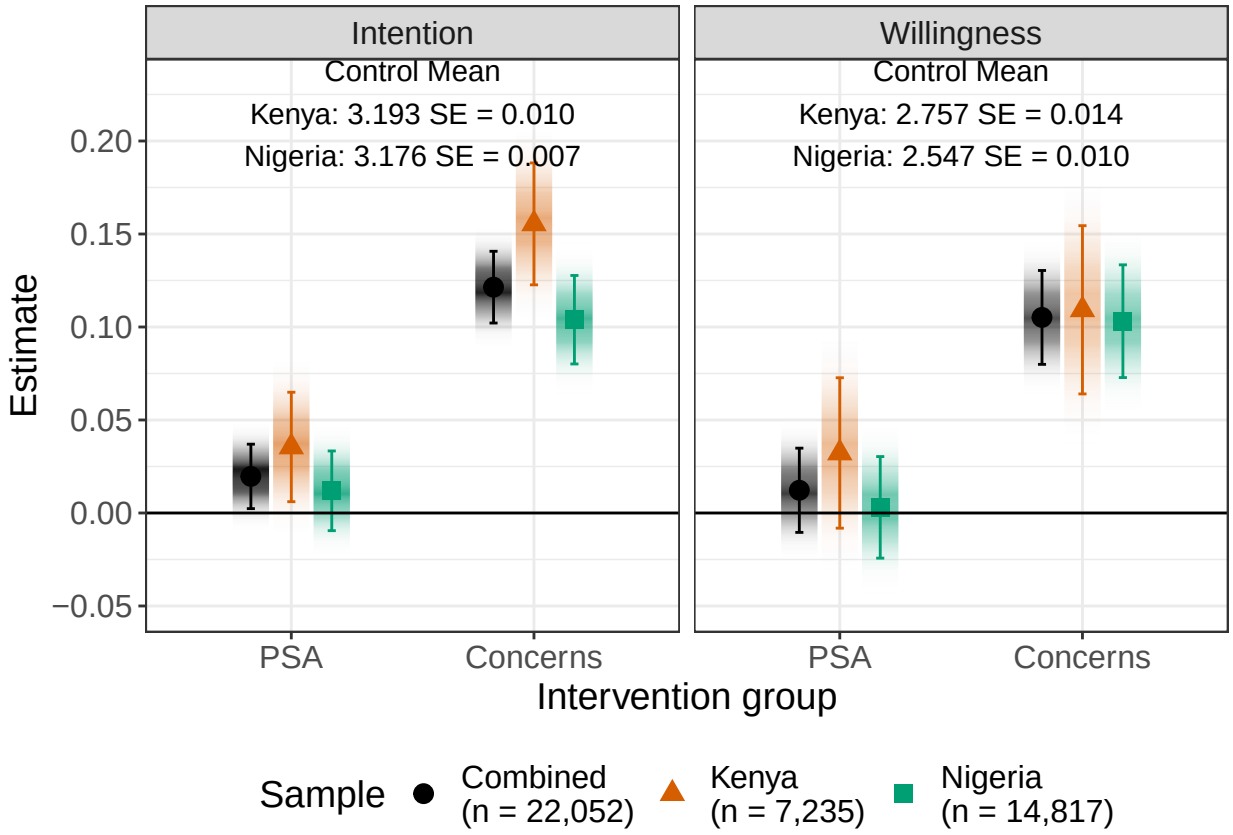
```
  mutate(treatment_group = relevel(treatment_group, ref = 'PSA'))
df_kenya_alt <- df_kenya |>
  mutate(treatment_group = relevel(treatment_group, ref = 'PSA'))
df_nigeria_alt <- df_nigeria |>
  mutate(treatment_group = relevel(treatment_group, ref = 'PSA'))

# intention dv
est_DM_intention <- lm_lin(get_vaccinated_2 ~ treatment_group,
                           data = df_eval, covariates = covariate_lin)
est_DM_intention_kenya <- lm_lin(get_vaccinated_2 ~ treatment_group,
                                 data = df_kenya, covariates = covariate_lin)
est_DM_intention_nigeria <- lm_lin(get_vaccinated_2 ~ treatment_group,
                                   data = df_nigeria,
                                   covariates = covariate_lin)

est_DM_intention_alt <- lm_lin(get_vaccinated_2 ~ treatment_group,
                               data = df_eval_alt,
                               covariates = covariate_lin)
est_DM_intention_kenya_alt <- lm_lin(get_vaccinated_2 ~ treatment_group,
                                     data = df_kenya_alt,
                                     covariates = covariate_lin)
est_DM_intention_nigeria_alt <- lm_lin(get_vaccinated_2 ~ treatment_group,
                                       data = df_nigeria_alt,
                                       covariates = covariate_lin)

group_ATE_get <- rbind.data.frame(
  cbind(est_DM_intention$coefficients[1:3],
        est_DM_intention$std.error[1:3],
        est_DM_intention$p.value[1:3]),
  cbind(est_DM_intention_kenya$coefficients[1:3],
        est_DM_intention_kenya$std.error[1:3],
        est_DM_intention_kenya$p.value[1:3]),
  cbind(est_DM_intention_nigeria$coefficients[1:3],
        est_DM_intention_nigeria$std.error[1:3],
        est_DM_intention_nigeria$p.value[1:3]))

names(group_ATE_get) <- c('coefficients','std.error','p.value')

group_ATE_get$`Intervention group` <- factor(
  levels(df_eval$treatment_group),
  levels = c('Control', 'PSA', 'Concerns'))
group_ATE_get$Sample <- rep(c('Combined','Kenya','Nigeria'), each = 3)

group_ATE_get$dv <- 'Intention'


# willingness dv
est_DM_willingness <- lm_lin(willingness_2 ~ treatment_group,
                             data = df_eval,
                             covariates = covariate_lin)
est_DM_willingness_kenya <- lm_lin(willingness_2 ~ treatment_group,
                                   data = df_kenya,
                                   covariates = covariate_lin)
```

```r
est_DM_willingness_nigeria <- lm_lin(willingness_2 ~ treatment_group,
                                     data = df_nigeria,
                                     covariates = covariate_lin)


est_DM_willingness_alt <- lm_lin(willingness_2 ~ treatment_group,
                                 data = df_eval_alt,
                                 covariates = covariate_lin)
est_DM_willingness_kenya_alt <- lm_lin(willingness_2 ~ treatment_group,
                                       data = df_kenya_alt,
                                       covariates = covariate_lin)
est_DM_willingness_nigeria_alt <- lm_lin(willingness_2 ~ treatment_group,
                                         data = df_nigeria_alt,
                                         covariates = covariate_lin)

group_ATE_will <- rbind.data.frame(
  cbind(est_DM_willingness$coefficients[1:3]
        ,est_DM_willingness$std.error[1:3],
        est_DM_willingness$p.value[1:3]),
  cbind(est_DM_willingness_kenya$coefficients[1:3],
        est_DM_willingness_kenya$std.error[1:3],
        est_DM_willingness_kenya$p.value[1:3]),
  cbind(est_DM_willingness_nigeria$coefficients[1:3],
        est_DM_willingness_nigeria$std.error[1:3],
        est_DM_willingness_nigeria$p.value[1:3]))

names(group_ATE_will) <- c('coefficients','std.error','p.value')

group_ATE_will$`Intervention group` <- factor(
  levels(df_eval$treatment_group),
  levels = c('Control', 'PSA', 'Concerns'))
group_ATE_will$Sample <- rep(c('Combined','Kenya','Nigeria'), each = 3)

group_ATE_will$dv <- 'Willingness'


# plot figure w/facets #
groupall <- rbind(group_ATE_get,group_ATE_will)
groupall$p <- round(groupall$p.value, digits = 3)

sample_sizes <- table(df_eval$Sample1)
#sum(sample_sizes)

gg_label <- filter(groupall, `Intervention group` == 'Control' &
                     Sample != 'Combined') |>
  mutate(label = paste0(Sample, ': ', sprintf('%.3f', coefficients),
                        ' SE = ', sprintf('%.3f', std.error))) |>
  group_by(dv) |>
  summarize(label = paste0(label, collapse = '\n'),
            Sample = 'Kenya')

ggplot(groupall[groupall$`Intervention group`!='Control',],
       aes(x=`Intervention group`, y = coefficients,
```

```
          color = Sample, shape = Sample)) +
stat_gradientinterval(aes(x = `Intervention group`,
                          ydist = distributional::dist_normal(coefficients,
                                                              std.error),
                          fill = Sample),
                      linewidth = 0,
                      point_size = 0,
                      point_alpha = 0,
                      interval_alpha = 0,
                      width = .5,
                      position = position_dodge(.5),
                      fill_type = 'segments') +

facet_wrap(~dv) +
geom_point(size=3, position = position_dodge(.5)) +
geom_errorbar(aes(ymin = coefficients - 1.96*std.error,
                  ymax = coefficients + 1.96*std.error),
              width = .1,
              position = position_dodge(.5)) +
scale_color_manual(values = c('black',cbPalette[7],cbPalette[4]),
                   labels = c(paste0('Combined\n(n = ',
                                     prettyNum(eval_n, big.mark = ','), ')'),
                              paste0('Kenya\n(n = ',
                                     prettyNum(kenya_n,
                                               big.mark = ','), ')'),
                              paste0('Nigeria\n(n = ',
                                     prettyNum(nigeria_n,
                                               big.mark = ','), ')')))+
scale_fill_manual(values = c('black',cbPalette[7],cbPalette[4]),
                  labels = c(paste0('Combined\n(n = ',
                                    prettyNum(eval_n, big.mark = ','), ')'),
                             paste0('Kenya\n(n = ',
                                    prettyNum(kenya_n,
                                              big.mark = ','), ')'),
                             paste0('Nigeria\n(n = ',
                                    prettyNum(nigeria_n,
                                              big.mark = ','), ')')),
                  guide = 'none') +
scale_shape_manual(values = c(19,17,15),
                   labels = c(paste0('Combined\n(n = ',
                                     prettyNum(eval_n, big.mark = ','), ')'),
                              paste0('Kenya\n(n = ',
                                     prettyNum(kenya_n,
                                               big.mark = ','), ')'),
                              paste0('Nigeria\n(n = ',
                                     prettyNum(nigeria_n,
                                               big.mark = ','), ')'))) +

geom_text(data = gg_label,
          aes(x = 1.5, y = .215, label = paste0('Control Mean\n', label)),
          color = 'black') +
ylab('Estimate') +
vcf_theme() +
```

```
  geom_hline(yintercept=0) +
  guides(shape = guide_legend(override.aes = list(linetype = 'blank'))) +
  scale_y_continuous(limits=c(-0.05, 0.23), breaks = seq(-0.05, 0.22, 0.05))
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```




```
intext_results <- groupall$coefficients[which(
  groupall$`Intervention group` == 'Concerns' &
    groupall$Sample == 'Combined')]/(
      groupall$coefficients[which(
        groupall$`Intervention group` == 'Control' &
          groupall$Sample == 'Combined')])

intext_results_kenya <- groupall$coefficients[which(
  groupall$`Intervention group` == 'Concerns' &
    groupall$Sample == 'Kenya')]/(
      groupall$coefficients[which(
        groupall$`Intervention group` == 'Control' &
          groupall$Sample == 'Kenya')])

intext_results_nigeria <- groupall$coefficients[which(
  groupall$`Intervention group` == 'Concerns' &
    groupall$Sample == 'Nigeria')]/(
      groupall$coefficients[which(
```

```
        groupall$`Intervention group` == 'Control' &
          groupall$Sample == 'Nigeria')])



##################################################
### code chunk number 4: hte
##################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| fig.align='center',
#| fig.height=5,
#| strip.white=TRUE,
#| results='asis'

# save counts
willingness_counts <- df_eval |>
  filter(willingness_group_post != 'Missing') |>
  mutate(group = willingness_group,
         outcome = 'willingness_2') |>
  group_by(group, outcome) |>
  summarize(count = n())
```

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

```
vaccinated_counts <- df_eval |>
  filter(get_vaccinated_group_post != 'Missing') |>
  mutate(group = get_vaccinated_group,
         outcome = 'get_vaccinated_2') |>
  group_by(group, outcome) |>
  summarize(count = n())
```

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

```
# regression: treatment interacted w/baseline group type
int.hes_w <- lm_robust(willingness_2 ~ treatment_group,
                       data = df_eval[df_eval$willingness_group=='Hesitant',])
int.und_w <- lm_robust(willingness_2 ~ treatment_group,
                       data = df_eval[df_eval$willingness_group=='Undecided',])
int.eag_w <- lm_robust(willingness_2 ~ treatment_group,
                       data = df_eval[df_eval$willingness_group=='Eager',])

int.hes_g <- lm_robust(get_vaccinated_2 ~ treatment_group,
                       data = df_eval[df_eval$get_vaccinated_group=='Hesitant',
                                      ])
int.und_g <- lm_robust(get_vaccinated_2 ~ treatment_group,
                       data = df_eval[df_eval$get_vaccinated_group=='Undecided',
                                      ])
int.eag_g <- lm_robust(get_vaccinated_2 ~ treatment_group,
                       data = df_eval[df_eval$get_vaccinated_group=='Eager',])

int.all <- bind_rows(Hesitant = tidy(int.hes_w)[-1,],
                     Undecided = tidy(int.und_w)[-1,],
                     Eager = tidy(int.eag_w)[-1,],
                     Hesitant = tidy(int.hes_g)[-1,],
```

```
                    Undecided = tidy(int.und_g)[-1,],
                    Eager = tidy(int.eag_g)[-1,],
                    .id = 'group')

int.all <- merge(int.all, rbind(willingness_counts,
                                vaccinated_counts),
                 by = c('group', 'outcome'), all.x = TRUE)

int.all <- int.all |>
  mutate(pre_test = paste0(group, ': ', prettyNum(count, big.mark = ',')),
         outcome = case_when(outcome == 'get_vaccinated_2' ~ 'Intention',
                             outcome == 'willingness_2' ~ 'Willingness'),
         term = factor(gsub('treatment_group', '', term),
                       levels = c('PSA', 'Concerns')
         ))


fig0 <- by(data = int.all, INDICES = int.all$outcome, FUN = function(m) {
  m <- droplevels(m)
  m$pre_test <- factor(m$pre_test, levels = sort(unique(m$pre_test))[c(2,3,1)])
  m <- ggplot(m,
              aes(x=term, y=estimate,
                  shape = pre_test,
                  color = pre_test)) +
    facet_grid(~outcome) +
    geom_point(size=3, position = position_dodge(.3)) +
    geom_errorbar(aes(ymin = conf.low,
                      ymax = conf.high),
                  width = .1,
                  position = position_dodge(.3)) +
    stat_gradientinterval(aes(x = term,
                              ydist = distributional::dist_normal(estimate,
                                                                  std.error),
                              fill = pre_test),
                          width = .5,
                          position = position_dodge(.3), .width = 0,
                          linewidth = 0,
                          fill_type = 'segments') +
    vcf_theme() +
    theme(legend.position = c(0.3, 0.8),
          axis.title.x=element_blank(),
          axis.title.y=element_blank(),
          legend.background=element_blank(),
          legend.title=element_blank(),
          legend.key = element_rect(fill = NA),
          legend.text=element_text(size=10),
          plot.background = element_rect(fill='transparent',
                                         color = 'transparent')) +
    scale_shape_manual(values = c(15, 17, 19),
                       name = 'Pre-test response') +
    scale_color_manual(values = c('red3','orange3','blue'),
                       name = 'Pre-test response') +
    scale_fill_manual(values = c('red3','orange3','blue'),
```

```
                      name = 'Pre-test response',
                      guide = 'none') +
    guides(shape = guide_legend(override.aes = list(linetype = 'blank'))) +
    geom_hline(yintercept=0) +
    coord_cartesian(ylim = c(-0.2, 0.6))
})
```

```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
#######################################################
### code chunk number 5: het_intent
#######################################################
#| eval = TRUE,
#| echo = FALSE, warning=FALSE, message=FALSE,
#| fig.align = 'center',
#| fig.height = 6,
#| fig.width = 13,
#| fig.pos = 'H',
#| prefix.string = 'fig',
#| fig.cap=
#| paste0('\\textbf{Composition of average treatment effects by pre-test
#|  vaccine intention.}',
#| ' The sample is users in the evaluation stage, $n = $ ',
#| prettyNum(eval_n, big.mark = ','),
#|  '. On the left panel, difference-in-means estimates of average treatment
#|  effects with respect to the control are grouped by pre-test response;
#|  error bars represent symmetric 95\\% confidence intervals with robust
#|  standard errors. Frequency of pre- and post-treatment response type by
#|  intervention group are reported in the right panel.'),
#| strip.white=TRUE,
#| results='asis'

# stack pre and post data frames
gg_levels <- data.frame(
  response = factor(c(df_eval$get_vaccinated_group,
                      df_eval$get_vaccinated_group_post),
                    levels = c('Missing','Eager','Undecided','Hesitant')),
  survey = factor(rep(c('Pre', 'Post'), each =  eval_n),
                  levels = c('Pre', 'Post')),
  group = df_eval$treatment_group,
  subject = rep(1:eval_n, times = 2))

agg_levels <- gg_levels |>
  group_by(survey, group, response) |>
  summarise(n = n()) |>
  mutate(perc = n / sum(n)) |>
  arrange(desc(response)) |>
  mutate(cumulative_n = case_when(perc < 0.03 ~ cumsum(n) + 150,
                                  TRUE ~ cumsum(n) - 0.5 * n))
```

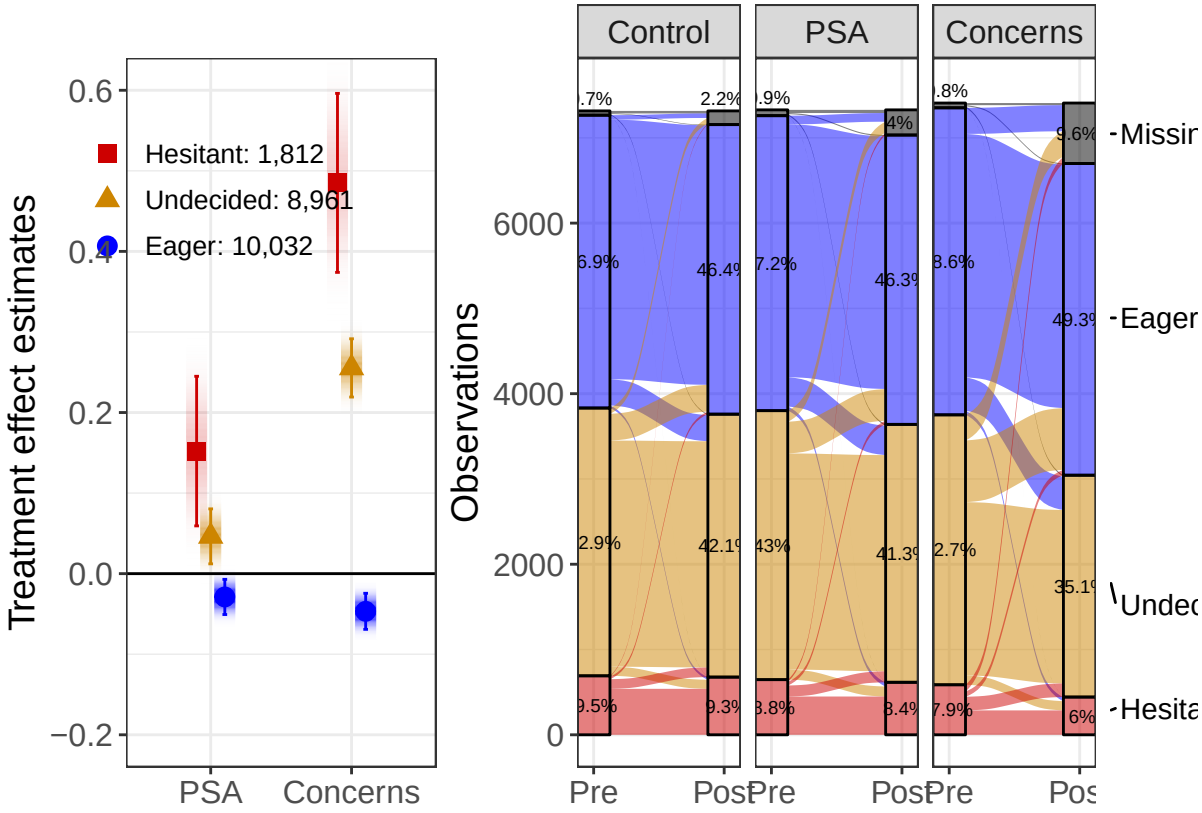
```
## `summarise()` has grouped output by 'survey', 'group'. You can override using the `.groups` argument
```

```r
agg_levels_pre_post <- gg_levels |>
  group_by(subject) |>
  mutate(pre_response = response[which(survey == 'Pre')],
         post_response = response[which(survey == 'Post')]) |>
  group_by(pre_response, post_response, group) |>
  summarise(n = n()) |>
  ungroup() |>
  group_by(group) |>
  mutate(perc = n / sum(n))
```

```
## `summarise()` has grouped output by 'pre_response', 'post_response'. You can override using the `.gr
## argument.
```

```r
fig_levels <- ggplot(gg_levels,
                     aes(x = survey, stratum = response, fill = response)) +
  facet_grid(~group) +
  scale_x_discrete(expand = c(.1, 0)) +
  geom_flow(aes(alluvium = subject), width = 1/4) +
  geom_stratum(aes(alluvium = subject), alpha = .5, width = 1/4) +
  geom_text(data = agg_levels,
            aes(y = cumulative_n, label = paste(round(perc*100,1), '%',sep='')),
            position = position_dodge(width = 0.05), size = 2.5) +
  scale_linetype_manual(values = c('blank', 'solid')) +
  scale_fill_manual(values = c('black', 'blue', 'orange3', 'red3')) +
  vcf_theme() +
  theme(legend.position = 'none',
        plot.margin = margin(0, 0, 0, 0, 'pt'),
        # no x-axis title
        axis.title.x = element_blank()) +
  ylab('Observations')


counts <- agg_levels |>
  filter(survey == 'Post' & group == 'Concerns')


labels <- ggplot(counts,
                 aes(x = 0, y = cumulative_n, label = response)) +
  ggrepel::geom_text_repel(min.segment.length = 0,
                           color = 'black',
                           size = 4,
                           nudge_x = 75
  ) +
  scale_x_continuous(limits = c(0, 100), expand = c(0, 0),
                     breaks = NULL, labels = NULL, name = NULL) +
  scale_y_continuous(limits = c(0, 7408), expand = c(0, 0),
                     breaks = NULL, labels = NULL, name = NULL) +
  theme(panel.background = element_blank(),
        plot.margin = margin(0, 0, 0, 0, 'pt'))


# add labels of categories to plot

p2 <- plot_spacer() + labels + plot_spacer() +
```

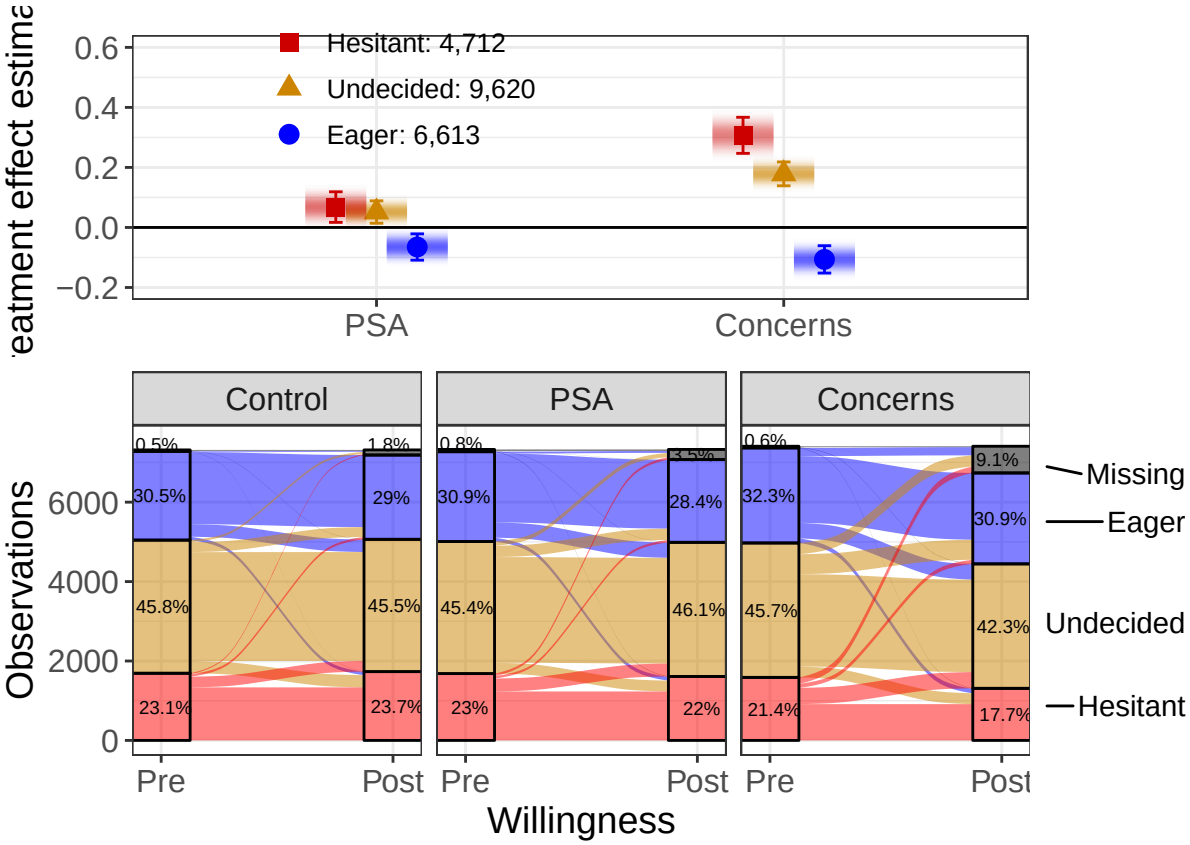
```
  plot_layout(heights = c(2,40,1.5))

fig1 <- fig_levels + p2 + plot_layout(widths = c(6,1))

fig01 <- fig0[[1]] +
  # remove header
  theme(strip.background = element_blank(),
        strip.text.x = element_blank(),
        axis.title.y = element_text()) +
  ylab('Treatment effect estimates')

# plot together
fig01 + fig1 + plot_layout(nrow = 1, ncol = 2, widths = c(1,2))
```




```
#####################################################
### code chunk number 6: het_willingness
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| fig.align='center',
#| fig.height=7,
#| results='hide',
#| include = FALSE

# stack pre and post data frames
gg_levels <- data.frame(
  response = factor(c(df_eval$willingness_group,
                      df_eval$willingness_group_post),
```

```
                    levels = c('Missing','Eager','Undecided','Hesitant')),
  survey = factor(rep(c('Pre', 'Post'), each =  eval_n),
                  levels = c('Pre', 'Post')),
  group = c(df_eval$treatment_group, df_eval$treatment_group),
  subject = rep(1:eval_n, times = 2))

agg_levels <- gg_levels |>
  group_by(survey, group, response) |>
  summarise(n = n()) |>
  mutate(perc = round(n / sum(n)*100,1)) |>
  arrange(desc(response)) |>
  mutate(cumulative_n = case_when(perc < 3 ~ cumsum(n) + 150,
                                  TRUE ~ cumsum(n) - 0.5 * n))
```

```
## `summarise()` has grouped output by 'survey', 'group'. You can override using the `.groups` argument
```

```
fig_levels <- ggplot(gg_levels,
                     aes(x = survey, stratum = response, fill = response)) +
  facet_grid(~group) +
  scale_x_discrete(expand = c(.1, 0)) +
  geom_flow(aes(alluvium = subject), width = 1/4) +
  geom_stratum(aes(alluvium = subject), alpha = .5, width = 1/4) +
  geom_text(data = agg_levels,
            aes(y = cumulative_n, label = paste(perc, '%',sep='')),
            position = position_dodge(width = 0.05), size = 2.5) +
  scale_linetype_manual(values = c('blank', 'solid')) +
  scale_fill_manual(values = c('black', 'blue', 'orange3', 'red')) +
  vcf_theme() +
  theme(legend.position = 'none',
        plot.margin = margin(0, 0, 0, 0, 'pt')) +
  xlab('Willingness') +
  ylab('Observations')


counts <- agg_levels |>
  filter(survey == 'Post' & group == 'Concerns')


labels <- ggplot(counts,
                 aes(x = 0, y = cumulative_n, label = response)) +
  ggrepel::geom_text_repel(min.segment.length = 0,
                           color = 'black',
                           size = 4,
                           nudge_x = 75
  ) +
  scale_x_continuous(limits = c(0, 100), expand = c(0, 0),
                     breaks = NULL, labels = NULL, name = NULL) +
  scale_y_continuous(limits = c(0, 7408), expand = c(0, 0),
                     breaks = NULL, labels = NULL, name = NULL) +
  theme(panel.background = element_blank(),
        plot.margin = margin(0, 0, 0, 0, 'pt'),
        # no x-axis title
        axis.title.x = element_blank())
```

```r
# add labels of categories to plot

p2 <- plot_spacer() + labels + plot_spacer() + plot_layout(heights = c(2,40,1.5))

fig2 <- fig_levels + p2 + plot_layout(widths = c(6,1))

fig02 <- fig0[[2]] +
  # remove header
  theme(strip.background = element_blank(),
        strip.text.x = element_blank(),
        axis.title.y = element_text()) +
  ylab('Treatment effect estimates') +
  plot_spacer() + plot_layout(widths = c(65,10))

# plot together
(fig02)/fig2 + plot_layout(heights = c(4,5))
```