chatbot_SI_jop.R

moffer

2024-11-29

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

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

require(knitr, quietly = TRUE)

options(modelsummary_factory_latex = 'kableExtra')

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

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



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

```
### 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')


## 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),
```

```
        # panel.border = element_rect(colour = 'gray50', fill=NA, size=.11),
        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')

covariate_list_full <- c(
  # pre-test response
  'willingness_f', 'get_vaccinated_f',
  # pre-registered covariates
  'is_male',
  'age', # no need for flag
  'education', 'education_flag',
  'is_urban',
  'religion_christian',
  'religion_muslim',
  'religion_pentecostal',
  'religiosity', 'religiosity_flag',
  'digital_index',
  'fb_post', 'fb_post_flag',
  'fb_msg', 'fb_msg_flag',
  'assets_index', 'assets_index_flag',
  'has_job',
  'hhold', 'hhold_flag',
  'party_aligned',
  'health_access',
  'cov_know',
  'cov_worried', 'cov_worried_flag',
  'cov_govt'
  #'cov_govt_flag'
)
# check distributions
# sapply(covariate_list_full, function(x){table(df_eval[,x], useNA = 'ifany') })

covariate_list <- covariate_list_full



##################################################
### code chunk number 3: balance
##################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'
```

```
covars <- c(
  #demographics
  'age', 'is_female', 'religion_christian', 'religiosity', 'party_aligned',
  #SES
  'is_urban', 'completed_secondary_school', 'hhold20', 'assets_index', 'has_job',
  #healthcare access, knowledge, attitudes
  'health_access',
  'cov_know',
  'know_wheretoget',
  'adult_vax',
  'getvax_easy',
  'trust_index')

df_treat <- df_eval |>
  mutate(is_female = 1*(is_male==0),
         hhold20 = case_when(hhold<20 ~hhold,
                             TRUE ~NA)) |>
  drop_na(treatment_group) |>
  group_by(treatment_group) |>
  summarise(across(all_of(covars),
                   .fns = list(mean = ~ sprintf('%.3f', mean(., na.rm = TRUE)),
                               se = ~ sprintf('(%.3f)',
                                              sd(., na.rm = TRUE)/sqrt(sum(!is.na(.))))),
                   .names = '{.fn}_{.col}'
  )) |> t()

row.names(df_treat)[grep('se_', row.names(df_treat))] <- ''

label_rows <- c('mean_age' = 'Age',
                'mean_is_female' = '% Non-male',
                'mean_health_access' = 'Hours to nearest health facility',
                'mean_party_aligned' = '% Party aligned',
                'mean_cov_know' = 'COVID knowledge index (-5:5)',
                'mean_religion_christian' = '% Christian',
                'mean_religiosity' = 'Religiosity (1:6)',
                'mean_completed_secondary_school' = '% Completed secondary school',
                'mean_is_urban' = '% Urban',
                'mean_hhold20' = 'Household size',
                'mean_assets_index' = 'Assets index (0:6)',
                'mean_has_job' = '% Employed',
                'mean_trust_index' = 'Trust index (-14:14)',
                'mean_adult_vax' = '% Any prior vaccination',
                'mean_know_wheretoget' = '% Know where to get COVID vaccine',
                'mean_getvax_easy' = 'Ease of getting COVID vaccine (-2:2)')

rownames(df_treat)[match(names(label_rows),
                         rownames(df_treat))] <- label_rows

colnames(df_treat) <- c("Control","PSA","Concerns")

kbl(df_treat[-1,],
    format = 'latex',
    caption= 'Balance table: covariate means and standard errors by intervention group',
```

```
  align = 'c', booktabs = TRUE,
  linesep = '',
  toprule = '\\vspace{-2.5em} \\\\ \\toprule') |>
kable_styling(latex_options = c('HOLD_position')) |>
pack_rows('Demographics', 2, 3) |>
pack_rows(NULL, 4, 5, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 6, 7, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 8, 9, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 10, 11, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 12, 13, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows('Socioeconomic status', 14, 15) |>
pack_rows(NULL, 16, 17, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 18, 19, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 20, 21, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 22, 23, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows('Healthcare access/knowledge/attitudes', 24, 25) |>
pack_rows(NULL, 26, 27, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 28, 29, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 30, 31, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 32, 33, bold = FALSE, latex_gap_space = '-1.85\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
footnote(general = paste0('\\\\footnotesize The sample is users in the evaluation stage, $n = $ ',
                          prettyNum(eval_n, big.mark = ','),
                          ". Estimates are of means grouped by treatment condition, produced from samp
         escape = FALSE,
         threeparttable = TRUE,
         general_title = '')
```

Table 1: Balance table: covariate means and standard errors by intervention group

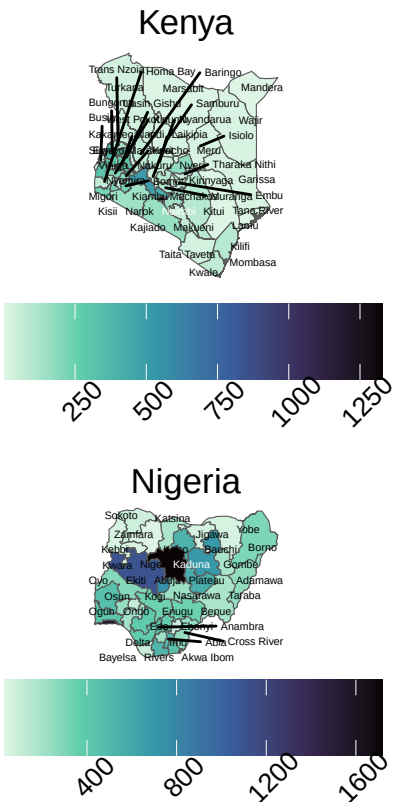
	Control	PSA	Concerns
Demographics			
Age	24.915	24.890	24.975
	(0.077)	(0.078)	(0.078)
% Non-male	0.471	0.487	0.472
	(0.006)	(0.006)	(0.006)
% Christian	0.775	0.785	0.781
	(0.005)	(0.005)	(0.005)
Religiosity (1:6)	3.920	3.914	3.937
	(0.019)	(0.019)	(0.019)
% Party aligned	0.298	0.301	0.302
	(0.005)	(0.005)	(0.005)
% Urban	0.571	0.582	0.575
	(0.006)	(0.006)	(0.006)
Socioeconomic status			
% Completed secondary school	0.911	0.911	0.912
	(0.003)	(0.003)	(0.003)
Household size	5.776	5.728	5.800
	(0.039)	(0.038)	(0.039)
Assets index (0:6)	4.288	4.288	4.255
	(0.017)	(0.017)	(0.017)
% Employed	0.359	0.355	0.359
	(0.006)	(0.006)	(0.006)
Hours to nearest health facility	1.805	1.816	1.727
	(0.048)	(0.047)	(0.046)
Healthcare access/knowledge/attitudes			
COVID knowledge index (-5:5)	3.676	3.689	3.697
	(0.013)	(0.013)	(0.013)
% Know where to get COVID vaccine	0.724	0.724	0.720
	(0.005)	(0.005)	(0.005)
% Any prior vaccination	0.406	0.406	0.406
	(0.006)	(0.006)	(0.006)
Ease of getting COVID vaccine (-2:2)	2.969	2.977	3.010
	(0.012)	(0.012)	(0.012)
Trust index (-14:14)	1.106	1.052	1.291
	(0.082)	(0.082)	(0.081)

The sample is users in the evaluation stage, n = 22,052. Estimates are of means grouped by treatment condition, produced from sample means and standard error of the sample mean.

```
#####################################################
### code chunk number 4: resp_map
#####################################################
#| eval = TRUE,
#| cache = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| fig.align='center',
#| fig.height=9,
#| fig.cap=
```

```r
#| paste0('\\textbf{Distribution of respondents across counties in Kenya and states in Nigeria.}'),
#| prefix.string = 'fig',
#| results='asis'


# get count of respondents by state (nigeria) / county (kenya)
count_nigeria_states <- as.data.frame(table(df_eval[df_eval$country1=="nigeria",]$region))
count_kenya_counties <- as.data.frame(table(df_eval[df_eval$country1=="kenya",]$region))

names(count_nigeria_states) <- names(count_kenya_counties) <- c("admin","count")

# get spatial data for nigeria and kenya
# downloaded from: https://data.humdata.org/dataset/cod-ab-nga?
nigeria_map <- read_sf("nga_adm_osgof_20190417/nga_admbnda_adm1_osgof_20190417.shp", quiet = TRUE, stri

# load kenya county shapefile (natural earth only has province level)
# downloaded from: https://gadm.org/download_country_v3.html#google_vignette
kenya_map <- read_sf("gadm36_KEN_shp/gadm36_KEN_1.shp", quiet = TRUE, stringsAsFactors = FALSE,as_tibbl

# clean kenya county punctutation
#sum(unique(kenya_map$NAME_1) %in% count_kenya_counties$admin)
#unique(kenya_map$NAME_1)[!unique(kenya_map$NAME_1) %in% count_kenya_counties$admin]

count_kenya_counties$admin <- sub("/"," ",as.character(count_kenya_counties$admin))
kenya_map$NAME_1 <- sub("'","",as.character(kenya_map$NAME_1))
kenya_map$NAME_1 <- sub("-"," ",as.character(kenya_map$NAME_1))

# change FCT to abuja
nigeria_map$ADM1_EN <- ifelse(nigeria_map$ADM1_EN == "Federal Capital Territory", "Abuja", nigeria_map$

# merge spatial and count data
nigeria_map <- merge(nigeria_map, count_nigeria_states, by.x="ADM1_EN", by.y="admin")
kenya_map <- merge(kenya_map, count_kenya_counties, by.x="NAME_1", by.y="admin")


# Plotting the heatmaps

# create threshold for white text labels
dark_threshold <- 1200

# Nigeria:
# Compute the centroids of each state for labeling
centroids <- st_centroid(st_geometry(nigeria_map))
nigeria_map$lon <- st_coordinates(centroids)[, 1]
nigeria_map$lat <- st_coordinates(centroids)[, 2]

nigeria_map$text_color <- ifelse(nigeria_map$count > dark_threshold, "white", "black")

nigeria_plot <- ggplot(data = nigeria_map) +
  geom_sf(aes(fill = count)) +
  geom_text_repel(aes(x = lon, y = lat, label = ADM1_EN, color = text_color), size = 1.5, box.padding =
  #geom_text(aes(x = lon, y = lat, label = name, color = text_color), size = 3, check_overlap = TRUE) +
  scale_color_identity() +  # This ensures colors are taken directly from the text_color column
```

```
  scale_fill_viridis_c(direction = -1, option = "mako", name = "") +
  theme_void() +
  labs(title = "Nigeria") +
  theme(
    axis.title = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank(),
    legend.position = "bottom",  # Position legend at the bottom
    legend.key.size = unit(1, "cm"),       # Increase legend key size
    legend.text = element_text(size = 10, angle = 45),  # Reduce font size and rotate
    plot.title = element_text(hjust = 0.5)  # Center the title
  )


# Kenya:
# Compute the centroids of each state for labeling
centroids <- st_centroid(st_geometry(kenya_map))
kenya_map$lon <- st_coordinates(centroids)[, 1]
kenya_map$lat <- st_coordinates(centroids)[, 2]

kenya_map$text_color <- ifelse(kenya_map$count > dark_threshold, "white", "black")

kenya_plot <- ggplot(data = kenya_map) +
  geom_sf(aes(fill = count)) +
  geom_text_repel(aes(x = lon, y = lat, label = NAME_1, color = text_color), size = 1.5, box.padding = 
  #geom_text(aes(x = lon, y = lat, label = NAME_1, color = text_color), size = 3, check_overlap = FALSE
  scale_color_identity() +
  scale_fill_viridis_c(
    direction = -1,
    option = "mako",
    name = "")+
  #breaks = seq(0, max(kenya_map$count), by = 100)) + # Adjust "by" based on your data range) +
  theme_void() +  # Use theme_void() for a cleaner look
  labs(
    title = "Kenya",
    #caption = "Source: Your Data Source"  # Adjust as needed
  ) +
  theme(
    axis.title = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank(),
    legend.position = "bottom",  # Position legend at the bottom
    legend.key.size = unit(1, "cm"),       # Increase legend key size
    legend.text = element_text(size = 10, angle = 45),  # Reduce font size and rotate
    plot.title = element_text(hjust = 0.5)  # Center the title
  )


# plot kenya and nigeria on top of each other
kenya_plot / nigeria_plot
```




```
#######################################################
### code chunk number 5: kenya_ab_table
#######################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'
df_afro_kenya_2019 <- read_sav("afrobarometer_kenya_2019.sav")

df_afro_kenya <- df_afro_kenya_2019 |>
  mutate(
    age = ifelse(Q1!=998|999|-1, Q1, NA),
    is_male = ifelse(Q101==1, 1, 0),
    is_christian = case_when(Q98A == 1|
                               Q98A == 2|
                               Q98A == 3|
                               Q98A == 4|
                               Q98A == 5|
                               Q98A == 6|
                               Q98A == 7|
                               Q98A == 8|
                               Q98A == 9|
                               Q98A == 10|
                               Q98A == 11|
                               Q98A == 12|
                               Q98A == 13|
                               Q98A == 14|
                               Q98A == 15|
```

```
                              Q98A == 16|
                              Q98A == 17|
                              Q98A == 30|
                              Q98A == 31|
                              Q98A == 32|
                              Q98A == 33 ~ 1,
                            Q98A == 0|
                              Q98A == 18|
                              Q98A == 19|
                              Q98A == 20|
                              Q98A == 21|
                              Q98A == 22|
                              Q98A == 23|
                              Q98A == 24|
                              Q98A == 25|
                              Q98A == 26|
                              Q98A == 27|
                              Q98A == 28|
                              Q98A == 29|
                              Q98A == 34|
                              Q98A == 300|
                              Q98A == 9995 ~ 0,
                            TRUE ~ NA_real_),
  is_urban = case_when(URBRUR == 1 ~ 1,
                       URBRUR == 2 ~ 0,
                       TRUE ~ NA_real_),
  completed_secondary_school = case_when(Q97 == 0|
                                           Q97 == 1|
                                           Q97 == 2|
                                           Q97 == 3|
                                           Q97 == 4 ~ 0,
                                         Q97 == 5|
                                           Q97 == 6|
                                           Q97 == 7|
                                           Q97 == 8|
                                           Q97 == 9 ~ 1,
                                         TRUE ~ NA_real_),
  # assets_index =
  #      # our assets: radio, tv, vehicle, computer, bank, phone (0-6) A-F in AB
  assets_index =
    case_when(Q92A == 0 ~ 0,
              Q92A == 1|
                Q92A == 2 ~ 1,
              TRUE ~ 0) +
    case_when(Q92B == 0 ~ 0,
              Q92B == 1|
                Q92B == 2 ~ 1,
              TRUE ~ 0) +
    case_when(Q92C == 0 ~ 0,
              Q92C == 1|
                Q92C == 2 ~ 1,
              TRUE ~ 0) +
    case_when(Q92D == 0 ~ 0,
```

```
                Q92D == 1|
                  Q92D == 2 ~ 1,
                TRUE ~ 0) +
      case_when(Q92E == 0 ~ 0,
                Q92E == 1|
                  Q92E == 2 ~ 1,
                TRUE ~ 0) +
      case_when(Q92F == 0 ~ 0,
                Q92F == 1|
                  Q92F == 2 ~ 1,
                TRUE ~ 0),
    has_job = case_when(Q95A == 0|
                          Q95A == 1 ~ 0,
                        Q95A == 2|
                          Q95A == 3 ~ 1,
                        TRUE ~ NA_real_),
    voted = case_when(Q13 == 0 ~ 0,
                      Q13 == 3 ~ 1,
                      TRUE ~ NA_real_),
    party_aligned = case_when(Q91B == 305 ~ 1,
                              Q91B != 305 ~ 0,
                              TRUE ~ NA_real_)) |>
  summarise(
    avg_age = as.numeric(sub("0+$", "", mean(age, na.rm=TRUE))),
    se_age = sd(age, na.rm = TRUE)/sqrt(sum(!is.na(age))),

    pct_female = 1 - mean(is_male, na.rm=TRUE),
    se_female = sd(is_male, na.rm = TRUE)/sqrt(sum(!is.na(is_male))),

    pct_christian = mean(is_christian, na.rm=TRUE),
    se_christian = sd(is_christian, na.rm = TRUE)/sqrt(sum(!is.na(is_christian))),

    pct_urban  = mean(is_urban, na.rm=TRUE),
    se_urban  = sd(is_urban, na.rm = TRUE)/sqrt(sum(!is.na(is_urban))),

    pct_completed_secondary_school = mean(completed_secondary_school, na.rm=TRUE),
    se_completed_secondary_school = sd(completed_secondary_school, na.rm = TRUE)/sqrt(sum(!is.na(comple

    avg_assets_index = mean(assets_index, na.rm = TRUE),
    se_assets_index = sd(assets_index, na.rm = TRUE)/sqrt(sum(!is.na(assets_index))),

    pct_has_job = mean(has_job, na.rm=TRUE),
    se_has_job = sd(has_job, na.rm = TRUE)/sqrt(sum(!is.na(has_job))),

    pct_voted = mean(voted, na.rm=TRUE),
    se_voted  = sd(voted, na.rm = TRUE)/sqrt(sum(!is.na(voted))),

    pct_party_aligned = mean(party_aligned, na.rm=TRUE),
    se_party_aligned  = sd(party_aligned, na.rm = TRUE)/sqrt(sum(!is.na(party_aligned)))
  ) |> t()


label_rows <- c('avg_age' = 'Age',
```

```
                'pct_female' = '% Non-male',
                'pct_christian' = '% Christian',
                'pct_urban' = '% Urban',
                'pct_completed_secondary_school' = '% Completed secondary school',
                'avg_assets_index' = 'Assets index (0:6)',
                'pct_has_job' = '% Employed',
                'pct_voted' = '% Voted last election',
                'pct_party_aligned' = '% Party aligned')

rownames(df_afro_kenya)[match(names(label_rows),
                              rownames(df_afro_kenya))] <- label_rows

# round to 3 digits
df_afro_kenya <- round(df_afro_kenya,3)

# add ( ) to se rows
df_afro_kenya[,1] <- ifelse(startsWith(rownames(df_afro_kenya), "se_"), paste0("(", round(df_afro_kenya

row.names(df_afro_kenya)[grep('se_', row.names(df_afro_kenya))] <- ''


## survey data - kenya ##

ab_covars <- c(
  #demographics
  'age', 'is_female', 'religion_christian',
  #SES
  'is_urban', 'completed_secondary_school', 'assets_index',
  'has_job',
  'voted','party_aligned')

df_dems <- df_eval |>
  filter(country1 == "kenya") |>
  mutate(is_female = 1*(is_male==0)) |>
  summarise(across(all_of(ab_covars),
                   .fns = list(mean = ~ sprintf('%.3f', mean(., na.rm = TRUE)),
                               se = ~ sprintf('(%.3f)',
                                              sd(., na.rm = TRUE)/sqrt(sum(!is.na(.))))),
                   .names = '{.fn}_{.col}'
  )) |> t()

# remove "se" from names
row.names(df_dems)[grep('se_', row.names(df_dems))] <- ''

label_rows <- c('mean_age' = 'Age',
                'mean_is_female' = '% Non-male',
                'mean_religion_christian' = '% Christian',
                'mean_is_urban' = '% Urban',
                'mean_completed_secondary_school' = '% Completed secondary school',
                'mean_assets_index' = 'Assets index (0:6)',
                'mean_has_job' = '% Employed',
                'mean_voted' = '% Voted last election',
                'mean_party_aligned' = '% Party aligned')
```

```r
rownames(df_dems)[match(names(label_rows),
                        rownames(df_dems))] <- label_rows


## combine data sets ##
kenya_fbab <- cbind(df_dems,df_afro_kenya)

colnames(kenya_fbab) <- c("Study sample","Afrobarometer (2019)")

## kenya table ##

kbl(kenya_fbab,
    format = 'latex',
    caption= 'Kenya sample characteristics compared to Afrobarometer',
    align = 'c', booktabs = TRUE,
    linesep = '',
    toprule = '\\vspace{-2.5em} \\\\ \\toprule') |>
  kable_styling(latex_options = c('HOLD_position'))  |>
  pack_rows(NULL, 2, 3, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 4, 5, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 6, 7, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 8, 9, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 10, 11, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 12, 13, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 14, 15, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 16, 17, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 18, 19, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  footnote(general = paste0('\\\\footnotesize The study sample is users in the evaluation stage, $n = $
                            prettyNum(eval_n, big.mark = ','),
                            ". Estimates are of means, produced from sample means and standard error of
           escape = FALSE,
           threeparttable = TRUE,
           general_title = '')
```

Table 2: Kenya sample characteristics compared to Afrobarometer

	Study sample	Afrobarometer (2019)
Age	25.620 (0.072)	36.338 (0.291)
% Non-male	0.479 (0.006)	0.5 (0.01)
% Christian	0.926 (0.003)	0.885 (0.007)
% Urban	0.414 (0.006)	0.357 (0.01)
% Completed secondary school	0.887 (0.004)	0.476 (0.01)
Assets index (0:6)	3.662 (0.017)	3.546 (0.03)
% Employed	0.284 (0.005)	0.48 (0.01)
% Voted last election	0.561 (0.006)	0.828 (0.008)
% Party aligned	0.358 (0.006)	0.254 (0.009)

The study sample is users in the evaluation stage, n = 22,052. Estimates are of means, produced from sample means and standard error of the sample mean.

```
#######################################################
### code chunk number 6: nigeria_ab_table
#######################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'
#|
## nigeria ##

# load afrobarometer data
df_afro_nigeria_2021 <- read_sav("afrobarometer_nigeria_2021.sav")



df_afro_nigeria <- df_afro_nigeria_2021 |>
  mutate(
    age = ifelse(Q1!=998|999|-1, Q1, NA),
    is_male = ifelse(Q101==1, 1, 0),
    is_christian = case_when(Q98A == 1|
                               Q98A == 2|
                               Q98A == 3|
                               Q98A == 4|
                               Q98A == 5|
                               Q98A == 6|
                               Q98A == 7|
                               Q98A == 8|
```

```
                              Q98A == 9|
                              Q98A == 10|
                              Q98A == 11|
                              Q98A == 12|
                              Q98A == 13|
                              Q98A == 14|
                              Q98A == 15|
                              Q98A == 16|
                              Q98A == 17|
                              Q98A == 30|
                              Q98A == 31|
                              Q98A == 32|
                              Q98A == 33 ~ 1,
                            Q98A == 0|
                              Q98A == 18|
                              Q98A == 19|
                              Q98A == 20|
                              Q98A == 21|
                              Q98A == 22|
                              Q98A == 23|
                              Q98A == 24|
                              Q98A == 25|
                              Q98A == 26|
                              Q98A == 27|
                              Q98A == 28|
                              Q98A == 29|
                              Q98A == 34|
                              Q98A == 300|
                              Q98A == 9995 ~ 0,
                            TRUE ~ NA_real_),
  is_urban = case_when(URBRUR == 1 ~ 1,
                       URBRUR == 2 ~ 0,
                       TRUE ~ NA_real_),
  completed_secondary_school = case_when(Q97 == 0|
                                           Q97 == 1|
                                           Q97 == 2|
                                           Q97 == 3|
                                           Q97 == 4 ~ 0,
                                         Q97 == 5|
                                           Q97 == 6|
                                           Q97 == 7|
                                           Q97 == 8|
                                           Q97 == 9 ~ 1,
                                         TRUE ~ NA_real_),
  assets_index =
    # our assets: radio, tv, vehicle, computer, bank, phone, **bike not asked in nigeria survey**
    case_when(Q92A == 0 ~ 0,
              Q92A == 1|
                Q92A == 2 ~ 1,
              TRUE ~ 0) +
    case_when(Q92B == 0 ~ 0,
              Q92B == 1|
                Q92B == 2 ~ 1,
```

```
              TRUE ~ 0) +
    case_when(Q92C == 0 ~ 0,
              Q92C == 1|
                Q92C == 2 ~ 1,
              TRUE ~ 0) +
    case_when(Q92D == 0 ~ 0,
              Q92D == 1|
                Q92D == 2 ~ 1,
              TRUE ~ 0) +
    case_when(Q92E == 0 ~ 0,
              Q92E == 1|
                Q92E == 2 ~ 1,
              TRUE ~ 0) +
    case_when(Q92F == 0 ~ 0,
              Q92F == 1|
                Q92F == 2 ~ 1,
              TRUE ~ 0),
  has_job = case_when(Q95A == 0|
                        Q95A == 1 ~ 0,
                      Q95A == 2|
                        Q95A == 3 ~ 1,
                      TRUE ~ NA_real_),
  voted = case_when(Q13 == 0 ~ 0,
                    Q13 == 3 ~ 1,
                    TRUE ~ NA_real_),
  party_aligned = case_when(Q91B == 621 ~ 1,
                            Q91B != 621 ~ 0,
                            TRUE ~ NA_real_))|>
summarise(
  avg_age = as.numeric(sub("0+$", "", mean(age, na.rm=TRUE))),
  se_age = sd(age, na.rm = TRUE)/sqrt(sum(!is.na(age))),

  pct_female = 1 - mean(is_male, na.rm=TRUE),
  se_female = sd(is_male, na.rm = TRUE)/sqrt(sum(!is.na(is_male))),

  pct_christian = mean(is_christian, na.rm=TRUE),
  se_christian = sd(is_christian, na.rm = TRUE)/sqrt(sum(!is.na(is_christian))),

  pct_urban  = mean(is_urban, na.rm=TRUE),
  se_urban  = sd(is_urban, na.rm = TRUE)/sqrt(sum(!is.na(is_urban))),

  pct_completed_secondary_school = mean(completed_secondary_school, na.rm=TRUE),
  se_completed_secondary_school = sd(completed_secondary_school, na.rm = TRUE)/sqrt(sum(!is.na(comple

  avg_assets_index = mean(assets_index, na.rm=TRUE),
  se_assets_index = sd(assets_index, na.rm = TRUE)/sqrt(sum(!is.na(assets_index))),

  pct_has_job = mean(has_job, na.rm=TRUE),
  se_has_job = sd(has_job, na.rm = TRUE)/sqrt(sum(!is.na(has_job))),

  pct_voted = mean(voted, na.rm=TRUE),
  se_voted  = sd(voted, na.rm = TRUE)/sqrt(sum(!is.na(voted))),
```

```
    pct_party_aligned = mean(party_aligned, na.rm=TRUE),
    se_party_aligned  = sd(party_aligned, na.rm = TRUE)/sqrt(sum(!is.na(party_aligned)))
  ) |> t()


label_rows <- c('avg_age' = 'Age',
                'pct_female' = '% Non-male',
                'pct_christian' = '% Christian',
                'pct_urban' = '% Urban',
                'pct_completed_secondary_school' = '% Completed secondary school',
                'avg_assets_index' = 'Assets index (0:6)',
                'pct_has_job' = '% Employed',
                'pct_voted' = '% Voted last election',
                'pct_party_aligned' = '% Party aligned')

rownames(df_afro_nigeria)[match(names(label_rows),
                                rownames(df_afro_nigeria))] <- label_rows

# round to 3 digits
df_afro_nigeria <- round(df_afro_nigeria,3)

# add ( ) to se rows
df_afro_nigeria[,1] <- ifelse(startsWith(rownames(df_afro_nigeria), "se_"), paste0("(", round(df_afro_ni

row.names(df_afro_nigeria)[grep('se_', row.names(df_afro_nigeria))] <- ''


## survey data - nigeria ##

ab_covars <- c(
  #demographics
  'age', 'is_female', 'religion_christian',
  #SES
  'is_urban', 'completed_secondary_school', 'assets_index',
  'has_job',
  'voted','party_aligned')

df_dems <- df_eval |>
  filter(country1 == "nigeria") |>
  mutate(is_female = 1*(is_male==0)) |>
  summarise(across(all_of(ab_covars),
                   .fns = list(mean = ~ sprintf('%.3f', mean(., na.rm = TRUE)),
                               se = ~ sprintf('(%.3f)',
                                              sd(., na.rm = TRUE)/sqrt(sum(!is.na(.))))),
                   .names = '{.fn}_{.col}'
  )) |> t()

# remove "se" from names
row.names(df_dems)[grep('se_', row.names(df_dems))] <- ''

label_rows <- c('mean_age' = 'Age',
                'mean_is_female' = '% Non-male',
                'mean_religion_christian' = '% Christian',
```

```
                'mean_is_urban' = '% Urban',
                'mean_completed_secondary_school' = '% Completed secondary school',
                'mean_assets_index' = 'Assets index (0:6)',
                'mean_has_job' = '% Employed',
                'mean_voted' = '% Voted last election',
                'mean_party_aligned' = '% Party aligned')

rownames(df_dems)[match(names(label_rows),
                        rownames(df_dems))] <- label_rows


## combine data sets ##
nigeria_fbab <- cbind(df_dems,df_afro_nigeria)

colnames(nigeria_fbab) <- c("Study sample","Afrobarometer (2021)")

## nigeria table ##

kbl(nigeria_fbab,
    format = 'latex',
    caption= 'Kenya sample characteristics compared to Afrobarometer',
    align = 'c', booktabs = TRUE,
    linesep = '',
    toprule = '\\vspace{-2.5em} \\\\ \\toprule') |>
  kable_styling(latex_options = c('HOLD_position'))  |>
  pack_rows(NULL, 2, 3, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 4, 5, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 6, 7, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 8, 9, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 10, 11, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 12, 13, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 14, 15, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 16, 17, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 18, 19, bold = FALSE, latex_gap_space = '-1.65\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  footnote(general = paste0('\\\\footnotesize The study sample is users in the evaluation stage, $n = $
                            prettyNum(eval_n, big.mark = ','),
                            ". Estimates are of means, produced from sample means and standard error of
           escape = FALSE,
           threeparttable = TRUE,
           general_title = '')
```

Table 3: Kenya sample characteristics compared to Afrobarometer

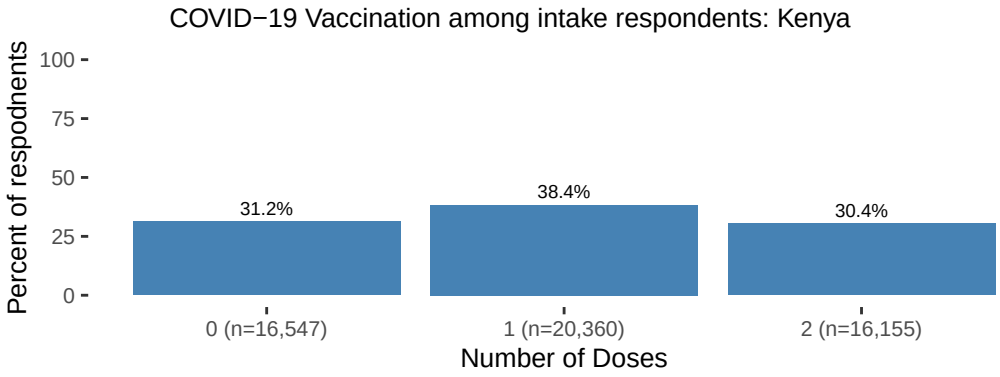
	Study sample	Afrobarometer (2021)
Age	24.589	34.769
	(0.056)	(0.686)
% Non-male	0.475	0.499
	(0.004)	(0.013)
% Christian	0.709	0.503
	(0.004)	(0.013)
% Urban	0.655	0.435
	(0.004)	(0.012)
% Completed secondary school	0.923	0.581
	(0.002)	(0.012)
Assets index (0:6)	4.577	3.684
	(0.011)	(0.04)
% Employed	0.393	0.461
	(0.004)	(0.013)
% Voted last election	0.448	0.792
	(0.004)	(0.01)
% Party aligned	0.272	0.295
	(0.004)	(0.011)

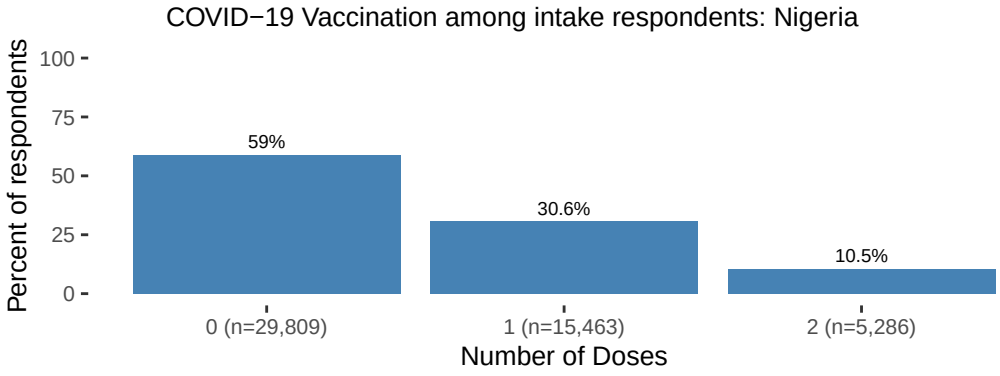
The study sample is users in the evaluation stage, $n = 22{,}052$. Estimates are of means, produced from sample means and standard error of the sample mean.

```
#####################################################
### code chunk number 7: intake_vaccines_calcs
#####################################################
#| eval = TRUE,
#| cache = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| reults = 'asis'

# load data

# create dataframes for figures
df <- read.csv("clean_vaxintake_data.csv")

vaccinations_kenya <- df |>
  filter(country1 == "kenya") |>
  drop_na(vax_status_numeric) |>
  mutate(count = 1,
         total_count = n()) |>
  group_by(vax_status_numeric) |>
  summarise(percent = round(sum(count / total_count) * 100, digits=1),
            count = paste(paste(" (n=", prettyNum(sum(count), big.mark = ','),sep=""),")",sep=""),sep="
  mutate(vax_status = paste(as.character(vax_status_numeric),count,sep="")) |>
  select(percent, vax_status)

vaccinations_nigeria <- df |>
  filter(country1 == "nigeria") |>
  drop_na(vax_status_numeric) |>
```

```
  mutate(count = 1,
         total_count = n()) |>
  group_by(vax_status_numeric) |>
  summarise(percent = round(sum(count / total_count) * 100, digits=1),
            count = paste(paste(" (n=", prettyNum(sum(count), big.mark = ','),sep=""),")",sep=""),sep="
  mutate(vax_status = paste(as.character(vax_status_numeric),count,sep="")) |>
  select(percent, vax_status)


####################################################
### code chunk number 8: intake_vaccines
####################################################
#| eval = TRUE,
#| cache = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| fig.align='center',
#| fig.height=5.5,
#| fig.cap=
#| paste0('\\textbf{Adaptive assignment in the learning stage.} ',
#| 'The algorithm updates separately within each concern category. ',
#| 'The sample is concerns expressed by users in the learning stage, $n = $ ',
#| prettyNum(nrow(df_concerns), big.mark = ','),
#| '; up to three concerns are accounted for from each of the ',
#| prettyNum(learn_n, big.mark = ','),
#| ' users in the learning stage.'),
#|
#| fig.cap=
#| paste0('\\textbf{Vaccine doses for all respondents at intake survey. }',
#| 'The sample is all users who completed the intake survey, $n = $ ',
#| prettyNum(sum(table(df$vax_status_numeric)), big.mark = ','),
#| '.'),
#| prefix.string = 'fig',
#| results='asis'

# kenya plot

in.kenya <- ggplot(vaccinations_kenya, aes(x=vax_status, y=percent)) +
  geom_bar(color="000000", fill="steelblue", stat="identity") +
  scale_y_continuous(limits = c(0, 100)) +
  geom_text(aes(label=paste0(percent,"%")), vjust=-.5, size=2.5) +
  xlab("Number of Doses") +
  ylab("Percent of respodnents") +
  ggtitle("COVID-19 Vaccination among intake respondents: Kenya") +
  theme(
    panel.background = element_blank(),
    plot.margin = margin(t = 10, r = 60, b = 10, l = 10),
    plot.title = element_text(size = 10, hjust = .3),
    text = element_text(size = 10))


# nigeria plot

in.nigeria <- ggplot(vaccinations_nigeria, aes(x=vax_status, y=percent)) +
```

```
  geom_bar(color="000000", fill="steelblue", stat="identity") +
  scale_y_continuous(limits = c(0, 100)) +
  geom_text(aes(label=paste0(percent,"%")), vjust=-.5, size=2.5) +
  xlab("Number of Doses") +
  ylab("Percent of respondents") +
  ggtitle("COVID-19 Vaccination among intake respondents: Nigeria") +
  theme(
    panel.background = element_blank(),
    plot.margin = margin(t = 10, r = 60, b = 10, l = 10),
    plot.title = element_text(size = 10, hjust = .3),
    text = element_text(size = 10))

in.kenya/in.nigeria
```






```
#########################################################
### code chunk number 9: joint_concerns_figure
#########################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| fig.align='center',
#| fig.width=12, fig.height=8,
#| prefix.string = 'fig',
#| fig.cap=
#| paste0('\\textbf{Adaptive assignment in the learning stage.} ',
#| 'The algorithm updates separately within each concern category. ',
#| 'The sample is concerns expressed by users in the learning stage, $n = $ ',
#| prettyNum(nrow(df_concerns), big.mark = ','),
#| '; up to three concerns are accounted for from each of the ',
#| prettyNum(learn_n, big.mark = ','),
```

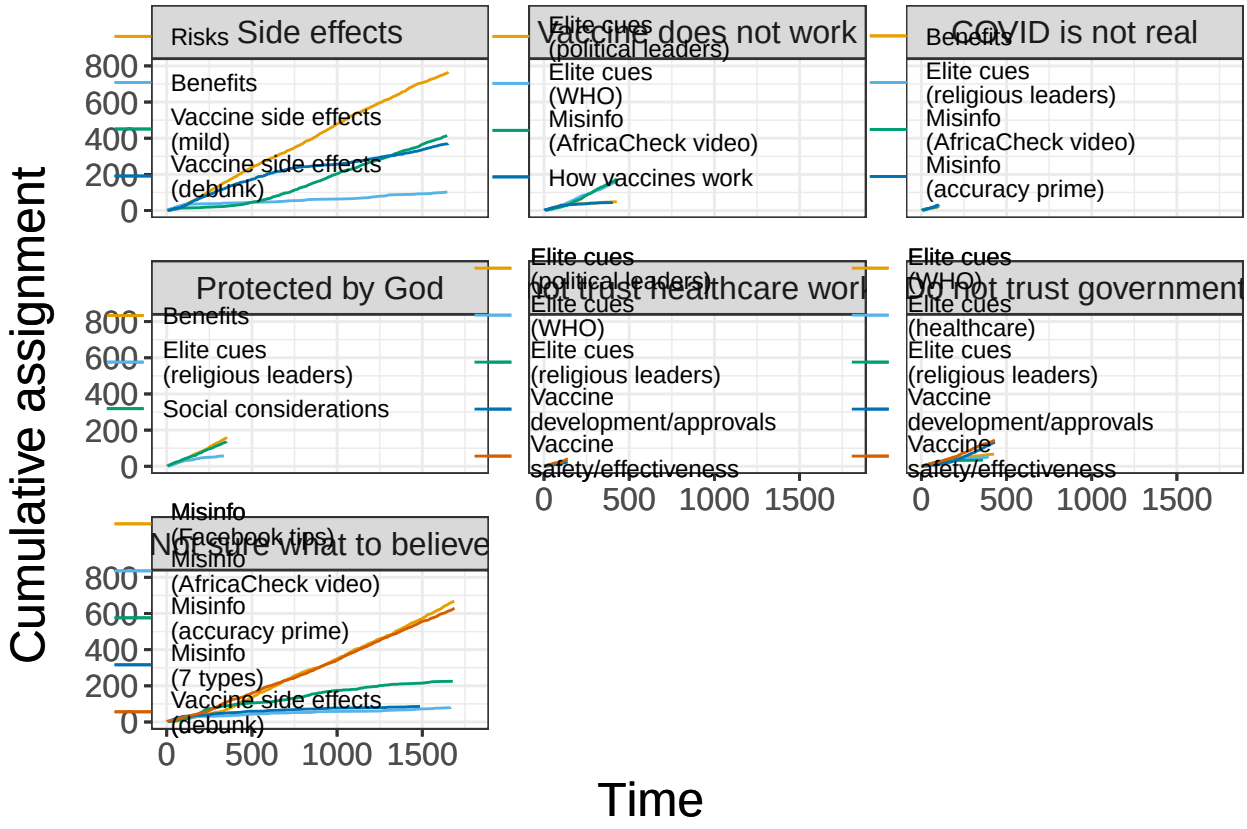
```
#| ' users in the learning stage.'),
#| strip.white=TRUE,
#| results='asis'
messages <- c(`1` = 'Risks',
              `2` = 'Benefits',
              `3` = 'Elite cues\n(political leaders)',
              `4` = 'Elite cues\n(WHO)',
              `5` = 'Elite cues\n(healthcare)',
              `6` = 'Elite cues\n(religious leaders)',
              `7` = 'Misinfo\n(Facebook tips)',
              `8` = 'Misinfo\n(AfricaCheck video)',
              `9` = 'Pledge',
              `10` = 'Misinfo\n(accuracy prime)',
              `11` = 'Misinfo\n(7 types)',
              `12` = 'Social considerations',
              `13` = 'How vaccines work',
              `14` = 'Vaccine\ndevelopment/approvals',
              `15` = 'Vaccine\nsafety/effectiveness',
              `16` = 'Vaccine side effects\n(mild)',
              `17` = 'Vaccine side effects\n(debunk)')

concernsl <- list(`1` = 'Side effects',
                  `2` = c('Vaccine does', 'not work'),
                  `3` = c('COVID is', 'not real'),
                  `4` = c('Protected', 'by God'),
                  `5` = c('Do not trust', 'healthcare', 'workers'),
                  `6` = c('Do not trust', 'government'),
                  `7` = c('Not sure what', 'to believe'))

concerns <- c(`1` = 'Side effects',
              `2` = 'Vaccine does not work',
              `3` = 'COVID is not real',
              `4` = 'Protected by God',
              `5` = 'Do not trust healthcare workers',
              `6` = 'Do not trust government',
              `7` = 'Not sure what to believe')

df_concerns$count <- ave(rep(1,nrow(df_concerns)), df_concerns$treatment_id,
                         df_concerns$concern_id, FUN = cumsum) # for graph
out <- list()
for(cid in sort(unique(df_concerns$concern_id)) ){
  ddfh <- df_concerns[which(df_concerns$concern_id == cid),]
  ddfh$time <- 1:nrow(ddfh)
  out[[cid]] <- ggplot(ddfh, aes(x = time, y = count, color = as.factor(treatment_id))) +
    geom_line() +
    facet_grid(~concern_id,
               labeller = labeller(concern_id = concerns)) +
    scale_color_manual(name = 'Message id', labels = as_labeller(messages),
                       values = cbPalette[-c(1, 5)]) +
    # ggtitle(concerns[cid]) +
    coord_cartesian(ylim = c(0, ceiling(max(df_concerns$count)/100)*100),
                    xlim = c(0, 1800)) +
    vcf_theme() +
```

```
    theme(legend.position = c(0.28,0.7),
          legend.background=element_blank(),
          legend.title=element_blank(),
          legend.key = element_rect(fill = NA),
          legend.text=element_text(size=8.5),
          axis.title.x=element_blank(),
          axis.title.y=element_blank(),
          plot.background = element_rect(fill='transparent',
                                         color = 'transparent'),
          plot.margin = margin(0, -10, 0, 0, "pt"))

  if(cid < 5){
    out[[cid]] <- out[[cid]] +
      theme(axis.text.x=element_text(colour = '#FFFFFF00'),
            axis.ticks.x=element_line(colour = '#FFFFFF00'))
  }

  if(cid %in% c(2,3,5,6)){
    out[[cid]] <- out[[cid]] +
      theme(axis.text.y=element_text(colour = "#FFFFFF00"),
            axis.ticks.y=element_line(colour = "#FFFFFF00"))
  }
}
```

```
## 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.
```

```
## 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.
```

```
bottom_text <- grid::textGrob('Time', gp = grid::gpar(fontsize = 18))
left_text <- grid::textGrob('Cumulative assignment', gp = grid::gpar(fontsize = 18), rot = 90, vjust =

g <- gridExtra::grid.arrange(grobs = out,
                             left = left_text,
                             bottom = bottom_text,
                             vp=grid::viewport(width=0.95, height=0.95))

grid::grid.draw(g)
```




```
#####################################################
### code chunk number 10: learning
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'
# Response estimates
hj_out <- by(df_concerns[, c('success', 'prob', 'treatment_id', 'concern_id')],
             list(treatment.concern =
                    interaction(df_concerns$treatment_id,
                                df_concerns$concern_id, drop = TRUE)),
             function(dd) {
               y <- dd$success
               probs <- dd$prob
               numer <- y/probs
               denom <- sum(1/probs)
               c(hj = sum(numer/denom),
                 sm = mean(y, na.rm = TRUE),
                 count = length(y), message = mean(dd$treatment_id), concern = mean(dd$concern_id))
             },
             simplify = FALSE)
hj_out <- data.frame(do.call(rbind, hj_out))
hj_out <- hj_out[order(hj_out[,'concern'], -hj_out[,'hj']),]

# primary result estimates for each treatment x concern consideration, hajek and sample mean

results_df <- data.frame(
```

```
  Messaging = messages[hj_out[,'message']],
  Concern = hj_out[,'concern'],
  LS = hj_out[,'count'], # learning sample
  SM = hj_out[,'sm'], # sample mean
  HJ = hj_out[,'hj'], # hajek estimate
  # 'inclusion error', decision rule for including second best arm,
  IE = sqrt(hj_out[,'hj']*(1-hj_out[,'hj']))*0.05,
  check.names = FALSE
)

results_df <- results_df |>
  group_by(Concern) |>
  arrange(-HJ, .by_group = TRUE) |>
  mutate(diff = HJ - first(HJ),
         group_id = seq_along(Concern),
         Selected = case_when((abs(diff) < first(IE) &
                                 group_id<=2) ~ 'Yes',
                              TRUE ~ 'No'),
         IE = case_when(group_id ==1 ~ IE,
                        TRUE ~ NA),
         # Allow concern names to fill multiple rows
         Concern = c(concernsl[[first(Concern)]],
                     rep('', length(Concern)-
                           length(concernsl[[first(Concern)]])))  ) |>
  ungroup() |>
  select(Concern, Messaging, LS, SM, HJ,
         IE, Selected)

kbl(results_df,
    caption= 'Learning from adaptive assignment.',
    col.names = linebreak(c('Concern', 'Messaging', 'Leaning\nsample',
                            'Sample\nmean', 'Hàjek\nestimate',
                            'Inclusion\nerror', 'Selected'),
                          align = c('l', 'r', 'c', 'c', 'c', 'c', 'c')),
    format = 'latex',
    digits = c(NA, NA, 0, 3,3,3,NA),
    toprule = '\\vspace{-3em} \\\\ \\toprule',
    align = 'lrrcccc', booktabs = TRUE,
    linesep = c( c( rep('', 3),  '\\addlinespace'),
                 c( rep('', 3),  '\\addlinespace'),
                 c( rep('', 3),  '\\addlinespace'),
                 c( rep('', 2),  '\\addlinespace'),
                 c( rep('', 4),  '\\addlinespace'),
                 c( rep('', 4),  '\\addlinespace'),
                 c( rep('', 4),  '\\addlinespace')),
    escape = FALSE)|>
  kable_styling(full_width = FALSE,
                latex_options = c('HOLD_position')) |>
  footnote(general = paste0('\\\\scriptsize The sample is concerns expressed by users in the learning s
                            prettyNum(nrow(df_concerns), big.mark = ','),'; up to three concerns are ac
  ),
  escape = FALSE,
  threeparttable = TRUE,
```

```
  general_title = '')
```

Table 4: Learning from adaptive assignment.

Concern	Messaging	Learning sample	Sample mean	Hàjek estimate	Inclusion error	Selected
Side effects	Risks	765	0.937	0.938	0.012	Yes
	Vaccine side effects (mild)	415	0.923	0.931		Yes
	Vaccine side effects (debunk)	371	0.916	0.930		No
	Benefits	103	0.883	0.928		No
Vaccine does not work	Misinfo (AfricaCheck video)	175	0.914	0.913	0.014	Yes
	Elite cues (political leaders)	49	0.776	0.902		Yes
	Elite cues (WHO)	159	0.855	0.862		No
	How vaccines work	46	0.761	0.755		No
COVID is not real	Misinfo (accuracy prime)	32	0.875	0.885	0.016	Yes
	Benefits	23	0.870	0.866		No
	Elite cues (religious leaders)	28	0.821	0.826		No
	Misinfo (AfricaCheck video)	22	0.727	0.727		No
Protected by God	Benefits	160	0.863	0.862	0.017	Yes
	Social considerations	137	0.839	0.839		No
	Elite cues (religious leaders)	57	0.737	0.748		No
Do not trust healthcare workers	Vaccine safety/effectiveness	41	0.976	0.985	0.006	Yes
	Vaccine development/approvals	24	0.917	0.916		No
	Elite cues (religious leaders)	26	0.885	0.890		No
	Elite cues (WHO)	23	0.870	0.878		No
	Elite cues (political leaders)	26	0.731	0.827		No
Do not trust government	Vaccine safety/effectiveness	146	0.897	0.902	0.015	Yes
	Vaccine development/approvals	133	0.872	0.876		No
	Elite cues (healthcare)	52	0.788	0.830		No
	Elite cues (WHO)	67	0.791	0.813		No
	Elite cues (religious leaders)	34	0.647	0.744		No
Not sure what to believe	Misinfo (Facebook tips)	668	0.936	0.936	0.012	Yes
	Vaccine side effects (debunk)	629	0.933	0.933		Yes
	Misinfo (accuracy prime)	225	0.898	0.912		No
	Misinfo (AfricaCheck video)	81	0.864	0.901		No
	Misinfo (7 types)	86	0.814	0.804		No

The sample is concerns expressed by users in the learning stage, $n = 4{,}803$; up to three concerns are accounted for from each of the 3,905 users in the learning stage. The "Learning sample" column represents the number of concerns addressed by a given messaging condition for a given concern, as assigned by the adaptive algorithm. The "Sample mean" and "Hàjek estimate" columns are estimates of mean response under counterfactual messaging, produced from sample means and a stabilized inverse probability weighted estimator; response is a binary measure of whether the respondent affirmed that the given messaging addressed their stated concern. The "Inclusion error" column is $0.05 \times \sqrt{\hat{p}(1-\hat{p})}$, where $\hat{p}$ is the Hàjek estimate of response. When the Hàjek estimate for the second best messaging is within the inclusion error of the best messaging, both options were selected to be randomly assigned in the evaluation phase, as pre-registered and reported in the "Selected" column.

```
#####################################################
### code chunk number 11: main_estimation
#####################################################
#| eval = TRUE,
#| cache = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'
```

```r
covariate_lin <- as.formula(paste('~' , paste0(covariate_list,
                                               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'))

# Response function General
## A) Difference-in-means estimate (i.e, treating missing data as random)
est_DM_vax <- estimatr::lm_robust(response ~ treatment_group, data = df_eval)
est_DM_vax_alt <- estimatr::lm_robust(response ~ treatment_group,
                                      data = df_eval_alt)

## B) Difference-in-means estimate , imputing pre-test as post-test
est_IMP_vax <- estimatr::lm_robust(response_imputed ~ treatment_group,
                                   data = df_eval)
est_IMP_vax_alt <- estimatr::lm_robust(response_imputed ~ treatment_group,
                                       data = df_eval_alt)


## C) Generalized random forest
est_GRF_vax <- mcf_estimate(df_eval, 'response', covariates = covariate_list)
est_GRF_vax_alt <- mcf_estimate(df_eval_alt, 'response',
                                covariates = covariate_list)

## D) IP Weighting
est_IP_vax <- IP_estimate(df_eval, 'response',
                          covariates = covariate_list)[['ATE']]
est_IP_vax_alt <- IP_estimate(df_eval_alt, 'response',
                              covariates = covariate_list)[['ATE']]

## E) Lin estimator
est_Lin_vax <- estimatr::lm_lin(response ~ treatment_group, data = df_eval,
                                covariates = covariate_lin)
est_Lin_vax_alt <- estimatr::lm_lin(response ~ treatment_group,
                                    data = df_eval_alt,
                                    covariates = covariate_lin)


# Response function Kenya
## A) Difference-in-means estimate (i.e, treating missing data as random)
est_DM_kenya <- estimatr::lm_robust(response ~ treatment_group, data = df_kenya)
est_DM_kenya_alt <- estimatr::lm_robust(response ~ treatment_group,
                                        data = df_kenya_alt)

## B) Difference-in-means estimate , imputing pre-test as post-test
est_IMP_kenya <- estimatr::lm_robust(response_imputed ~ treatment_group,
                                     data = df_kenya)
est_IMP_kenya_alt <- estimatr::lm_robust(response_imputed ~ treatment_group,
                                         data = df_kenya_alt)
```

```r
## C) Generalized random forest
est_GRF_kenya <- mcf_estimate(df_kenya, 'response', covariates = covariate_list)
est_GRF_kenya_alt <- mcf_estimate(df_kenya_alt, 'response',
                                  covariates = covariate_list)

## D) IP Weighting
est_IP_kenya <- IP_estimate(df_kenya, 'response',
                            covariates = covariate_list)[['ATE']]
est_IP_kenya_alt <- IP_estimate(df_kenya_alt, 'response',
                                covariates = covariate_list)[['ATE']]

## E) Lin estimator
est_Lin_kenya <- estimatr::lm_lin(response ~ treatment_group, data = df_kenya,
                                  covariates = covariate_lin)
est_Lin_kenya_alt <- estimatr::lm_lin(response ~ treatment_group,
                                      data = df_kenya_alt,
                                      covariates = covariate_lin)


# Response function Nigeria
## A) Difference-in-means estimate (i.e, treating missing data as random)
est_DM_nigeria <- estimatr::lm_robust(response ~ treatment_group, data = df_nigeria)
est_DM_nigeria_alt <- estimatr::lm_robust(response ~ treatment_group,
                                          data = df_nigeria_alt)

## B) Difference-in-means estimate , imputing pre-test as post-test
est_IMP_nigeria <- estimatr::lm_robust(response_imputed ~ treatment_group,
                                       data = df_nigeria)
est_IMP_nigeria_alt <- estimatr::lm_robust(response_imputed ~ treatment_group,
                                           data = df_nigeria_alt)

## C) Generalized random forest
est_GRF_nigeria <- mcf_estimate(df_nigeria, 'response', covariates = covariate_list)
est_GRF_nigeria_alt <- mcf_estimate(df_nigeria_alt, 'response',
                                    covariates = covariate_list)

## D) IP Weighting
est_IP_nigeria <- IP_estimate(df_nigeria, 'response',
                              covariates = covariate_list)[['ATE']]
est_IP_nigeria_alt <- IP_estimate(df_nigeria_alt, 'response',
                                  covariates = covariate_list)[['ATE']]

## E) Lin estimator
est_Lin_nigeria <- estimatr::lm_lin(response ~ treatment_group, data = df_nigeria,
                                    covariates = covariate_lin)
est_Lin_nigeria_alt <- estimatr::lm_lin(response ~ treatment_group,
                                        data = df_nigeria_alt,
                                        covariates = covariate_lin)


##################################################
### code chunk number 12: intention_estimation
##################################################
```

```
#| eval = TRUE,
#| cache = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

# Response function General
## A) Difference-in-means estimate (i.e, treating missing data as random)
est_DM_vax_i <- estimatr::lm_robust(get_vaccinated_2 ~ treatment_group, data = df_eval)
est_DM_vax_alt_i <- estimatr::lm_robust(get_vaccinated_2 ~ treatment_group,
                                        data = df_eval_alt)

## B) Difference-in-means estimate , imputing pre-test as post-test
est_IMP_vax_i <- estimatr::lm_robust(get_vaccinated_2_imputed ~ treatment_group,
                                     data = df_eval)
est_IMP_vax_alt_i <- estimatr::lm_robust(get_vaccinated_2_imputed ~ treatment_group,
                                         data = df_eval_alt)


## C) Generalized random forest
est_GRF_vax_i <- mcf_estimate(df_eval, 'get_vaccinated_2', covariates = covariate_list)
est_GRF_vax_alt_i <- mcf_estimate(df_eval_alt, 'get_vaccinated_2',
                                  covariates = covariate_list)

## D) IP Weighting
est_IP_vax_i <- IP_estimate(df_eval, 'get_vaccinated_2',
                            covariates = covariate_list)[['ATE']]
est_IP_vax_alt_i <- IP_estimate(df_eval_alt, 'get_vaccinated_2',
                                covariates = covariate_list)[['ATE']]

## E) Lin estimator
est_Lin_vax_i <- estimatr::lm_lin(get_vaccinated_2 ~ treatment_group, data = df_eval,
                                  covariates = covariate_lin)
est_Lin_vax_alt_i <- estimatr::lm_lin(get_vaccinated_2 ~ treatment_group,
                                      data = df_eval_alt,
                                      covariates = covariate_lin)


# Response function Kenya
## A) Difference-in-means estimate (i.e, treating missing data as random)
est_DM_kenya_i <- estimatr::lm_robust(get_vaccinated_2 ~ treatment_group, data = df_kenya)
est_DM_kenya_alt_i <- estimatr::lm_robust(get_vaccinated_2 ~ treatment_group,
                                          data = df_kenya_alt)

## B) Difference-in-means estimate , imputing pre-test as post-test
est_IMP_kenya_i <- estimatr::lm_robust(get_vaccinated_2_imputed ~ treatment_group,
                                       data = df_kenya)
est_IMP_kenya_alt_i <- estimatr::lm_robust(get_vaccinated_2_imputed ~ treatment_group,
                                           data = df_kenya_alt)

## C) Generalized random forest
est_GRF_kenya_i <- mcf_estimate(df_kenya, 'get_vaccinated_2', covariates = covariate_list)
est_GRF_kenya_alt_i <- mcf_estimate(df_kenya_alt, 'get_vaccinated_2',
```

```
                                    covariates = covariate_list)

## D) IP Weighting
est_IP_kenya_i <- IP_estimate(df_kenya, 'get_vaccinated_2',
                              covariates = covariate_list)[['ATE']]
est_IP_kenya_alt_i <- IP_estimate(df_kenya_alt, 'get_vaccinated_2',
                                  covariates = covariate_list)[['ATE']]

## E) Lin estimator
est_Lin_kenya_i <- estimatr::lm_lin(get_vaccinated_2 ~ treatment_group, data = df_kenya,
                                    covariates = covariate_lin)
est_Lin_kenya_alt_i <- estimatr::lm_lin(get_vaccinated_2 ~ treatment_group,
                                        data = df_kenya_alt,
                                        covariates = covariate_lin)


# Response function Nigeria
## A) Difference-in-means estimate (i.e, treating missing data as random)
est_DM_nigeria_i <- estimatr::lm_robust(get_vaccinated_2 ~ treatment_group, data = df_nigeria)
est_DM_nigeria_alt_i <- estimatr::lm_robust(get_vaccinated_2 ~ treatment_group,
                                            data = df_nigeria_alt)

## B) Difference-in-means estimate , imputing pre-test as post-test
est_IMP_nigeria_i <- estimatr::lm_robust(get_vaccinated_2_imputed ~ treatment_group,
                                         data = df_nigeria)
est_IMP_nigeria_alt_i <- estimatr::lm_robust(get_vaccinated_2_imputed ~ treatment_group,
                                             data = df_nigeria_alt)

## C) Generalized random forest
est_GRF_nigeria_i <- mcf_estimate(df_nigeria, 'get_vaccinated_2', covariates = covariate_list)
est_GRF_nigeria_alt_i <- mcf_estimate(df_nigeria_alt, 'get_vaccinated_2',
                                      covariates = covariate_list)

## D) IP Weighting
est_IP_nigeria_i <- IP_estimate(df_nigeria, 'get_vaccinated_2',
                                covariates = covariate_list)[['ATE']]
est_IP_nigeria_alt_i <- IP_estimate(df_nigeria_alt, 'get_vaccinated_2',
                                    covariates = covariate_list)[['ATE']]

## E) Lin estimator
est_Lin_nigeria_i <- estimatr::lm_lin(get_vaccinated_2 ~ treatment_group, data = df_nigeria,
                                      covariates = covariate_lin)
est_Lin_nigeria_alt_i <- estimatr::lm_lin(get_vaccinated_2 ~ treatment_group,
                                          data = df_nigeria_alt,
                                          covariates = covariate_lin)


#####################################################
### code chunk number 13: willing_estimation
#####################################################
#| eval = TRUE,
#| cache = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
```

```
#| strip.white=TRUE,
#| results='asis'

# Response function General
## A) Difference-in-means estimate (i.e, treating missing data as random)
est_DM_vax_w <- estimatr::lm_robust(willingness_2 ~ treatment_group, data = df_eval)
est_DM_vax_alt_w <- estimatr::lm_robust(willingness_2 ~ treatment_group,
                                        data = df_eval_alt)

## B) Difference-in-means estimate , imputing pre-test as post-test
est_IMP_vax_w <- estimatr::lm_robust(willingness_2_imputed ~ treatment_group,
                                     data = df_eval)
est_IMP_vax_alt_w <- estimatr::lm_robust(willingness_2_imputed ~ treatment_group,
                                         data = df_eval_alt)


## C) Generalized random forest
est_GRF_vax_w <- mcf_estimate(df_eval, 'willingness_2', covariates = covariate_list)
est_GRF_vax_alt_w <- mcf_estimate(df_eval_alt, 'willingness_2',
                                  covariates = covariate_list)

## D) IP Weighting
est_IP_vax_w <- IP_estimate(df_eval, 'willingness_2',
                            covariates = covariate_list)[['ATE']]
est_IP_vax_alt_w <- IP_estimate(df_eval_alt, 'willingness_2',
                                covariates = covariate_list)[['ATE']]

## E) Lin estimator
est_Lin_vax_w <- estimatr::lm_lin(willingness_2 ~ treatment_group, data = df_eval,
                                  covariates = covariate_lin)
est_Lin_vax_alt_w <- estimatr::lm_lin(willingness_2 ~ treatment_group,
                                      data = df_eval_alt,
                                      covariates = covariate_lin)


# Response function Kenya
## A) Difference-in-means estimate (i.e, treating missing data as random)
est_DM_kenya_w <- estimatr::lm_robust(willingness_2 ~ treatment_group, data = df_kenya)
est_DM_kenya_alt_w <- estimatr::lm_robust(willingness_2 ~ treatment_group,
                                          data = df_kenya_alt)

## B) Difference-in-means estimate , imputing pre-test as post-test
est_IMP_kenya_w <- estimatr::lm_robust(willingness_2_imputed ~ treatment_group,
                                       data = df_kenya)
est_IMP_kenya_alt_w <- estimatr::lm_robust(willingness_2_imputed ~ treatment_group,
                                           data = df_kenya_alt)

## C) Generalized random forest
est_GRF_kenya_w <- mcf_estimate(df_kenya, 'willingness_2', covariates = covariate_list)
est_GRF_kenya_alt_w <- mcf_estimate(df_kenya_alt, 'willingness_2',
                                    covariates = covariate_list)

## D) IP Weighting
```

```r
est_IP_kenya_w <- IP_estimate(df_kenya, 'willingness_2',
                              covariates = covariate_list)[['ATE']]
est_IP_kenya_alt_w <- IP_estimate(df_kenya_alt, 'willingness_2',
                                  covariates = covariate_list)[['ATE']]

## E) Lin estimator
est_Lin_kenya_w <- estimatr::lm_lin(willingness_2 ~ treatment_group, data = df_kenya,
                                    covariates = covariate_lin)
est_Lin_kenya_alt_w <- estimatr::lm_lin(willingness_2 ~ treatment_group,
                                        data = df_kenya_alt,
                                        covariates = covariate_lin)


# Response function Nigeria
## A) Difference-in-means estimate (i.e, treating missing data as random)
est_DM_nigeria_w <- estimatr::lm_robust(willingness_2 ~ treatment_group, data = df_nigeria)
est_DM_nigeria_alt_w <- estimatr::lm_robust(willingness_2 ~ treatment_group,
                                            data = df_nigeria_alt)

## B) Difference-in-means estimate , imputing pre-test as post-test
est_IMP_nigeria_w <- estimatr::lm_robust(willingness_2_imputed ~ treatment_group,
                                         data = df_nigeria)
est_IMP_nigeria_alt_w <- estimatr::lm_robust(willingness_2_imputed ~ treatment_group,
                                             data = df_nigeria_alt)

## C) Generalized random forest
est_GRF_nigeria_w <- mcf_estimate(df_nigeria, 'willingness_2', covariates = covariate_list)
est_GRF_nigeria_alt_w <- mcf_estimate(df_nigeria_alt, 'willingness_2',
                                      covariates = covariate_list)

## D) IP Weighting
est_IP_nigeria_w <- IP_estimate(df_nigeria, 'willingness_2',
                                covariates = covariate_list)[['ATE']]
est_IP_nigeria_alt_w <- IP_estimate(df_nigeria_alt, 'willingness_2',
                                    covariates = covariate_list)[['ATE']]

## E) Lin estimator
est_Lin_nigeria_w <- estimatr::lm_lin(willingness_2 ~ treatment_group, data = df_nigeria,
                                      covariates = covariate_lin)
est_Lin_nigeria_alt_w <- estimatr::lm_lin(willingness_2 ~ treatment_group,
                                          data = df_nigeria_alt,
                                          covariates = covariate_lin)


#####################################################
### code chunk number 14: main_results_intention
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

# Combine model objects
```

```
out_list_i <- list(
  DM = est_DM_vax_i,
  IMP = est_IMP_vax_i,
  Lin = est_Lin_vax_i,
  IP = est_IP_vax_i
)

out_list_kenya_i <- list(
  DM = est_DM_kenya_i,
  IMP = est_IMP_kenya_i,
  Lin = est_Lin_kenya_i,
  IP = est_IP_kenya_i
)

out_list_nigeria_i <- list(
  DM = est_DM_nigeria_i,
  IMP = est_IMP_nigeria_i,
  Lin = est_Lin_nigeria_i,
  IP = est_IP_nigeria_i
)

out_list_i <- modelsummary(out_list_i,
                           output = 'modelsummary_list')
out_list_kenya_i <- modelsummary(out_list_kenya_i,
                                 output = 'modelsummary_list')
out_list_nigeria_i <- modelsummary(out_list_nigeria_i,
                                   output = 'modelsummary_list')

for(x in names(out_list_i)){
  # add on Concerns - PSA coefficients
  out_list_i[[x]]$tidy <- rbind(out_list_i[[x]]$tidy |>
                                  select(term, estimate, std.error, statistic,
                                         p.value),
                                broom::tidy(get(paste0('est_', x, '_vax_alt_i'))) |>
                                  filter(term == 'treatment_groupConcerns') |>
                                  mutate(term = 'treatment_groupConcerns2') |>
                                  select(term, estimate, std.error, statistic,
                                         p.value))

  out_list_kenya_i[[x]]$tidy <- rbind(out_list_kenya_i[[x]]$tidy |>
                                        select(term, estimate, std.error, statistic,
                                               p.value),
                                      broom::tidy(get(paste0('est_', x, '_kenya_alt_i'))) |>
                                        filter(term == 'treatment_groupConcerns') |>
                                        mutate(term = 'treatment_groupConcerns2') |>
                                        select(term, estimate, std.error, statistic,
                                               p.value))

  out_list_nigeria_i[[x]]$tidy <- rbind(out_list_nigeria_i[[x]]$tidy |>
                                          select(term, estimate, std.error, statistic,
                                                 p.value),
                                        broom::tidy(get(paste0('est_', x, '_nigeria_alt_i'))) |>
                                          filter(term == 'treatment_groupConcerns') |>
```

```
                                        mutate(term = 'treatment_groupConcerns2') |>
                                        select(term, estimate, std.error, statistic,
                                               p.value))

  # drop p-values for control condition so no stars
  out_list_i[[x]]$tidy[which(out_list_i[[x]]$tidy$term == '(Intercept)'),
                       c('p.value', 'statistic')] <- NA
  out_list_kenya_i[[x]]$tidy[which(out_list_kenya_i[[x]]$tidy$term == '(Intercept)'),
                             c('p.value', 'statistic')] <- NA
  out_list_nigeria_i[[x]]$tidy[which(out_list_nigeria_i[[x]]$tidy$term == '(Intercept)'),
                               c('p.value', 'statistic')] <- NA
}

# Add GRF; different format
out_list_i[['GRF']] <- format_grf_summary(est_GRF_vax_i)
out_list_i[['GRF']]$tidy <- rbind(out_list_i[['GRF']]$tidy,
                                  format_grf_summary(est_GRF_vax_alt_i)$tidy |>
                                    filter(term == 'treatment_groupConcerns') |>
                                    mutate(term = 'treatment_groupConcerns2'))
out_list_i[['GRF']]$tidy[which(out_list_i[['GRF']]$tidy$term == '(Intercept)'),
                         c('p.value', 'statistic')] <- NA


out_list_kenya_i[['GRF']] <- format_grf_summary(est_GRF_kenya_i)
out_list_kenya_i[['GRF']]$tidy <- rbind(out_list_kenya_i[['GRF']]$tidy,
                                        format_grf_summary(est_GRF_kenya_alt_i)$tidy |>
                                          filter(term == 'treatment_groupConcerns') |>
                                          mutate(term = 'treatment_groupConcerns2'))
out_list_kenya_i[['GRF']]$tidy[which(out_list_kenya_i[['GRF']]$tidy$term == '(Intercept)'),
                               c('p.value', 'statistic')] <- NA

out_list_nigeria_i[['GRF']] <- format_grf_summary(est_GRF_nigeria_i)
out_list_nigeria_i[['GRF']]$tidy <- rbind(out_list_nigeria_i[['GRF']]$tidy,
                                          format_grf_summary(est_GRF_nigeria_alt_i)$tidy |>
                                            filter(term == 'treatment_groupConcerns') |>
                                            mutate(term = 'treatment_groupConcerns2'))
out_list_nigeria_i[['GRF']]$tidy[which(out_list_nigeria_i[['GRF']]$tidy$term == '(Intercept)'),
                                 c('p.value', 'statistic')] <- NA


modelsummary(list(Combined = out_list_i,
                  Kenya = out_list_kenya_i,
                  Nigeria = out_list_nigeria_i),
             shape= 'rbind',
             output = 'latex',
             coef_map = c('treatment_groupPSA' = 'PSA - Control',
                          'treatment_groupConcerns' = 'Concerns - Control',
                          'treatment_groupConcerns2' = 'Concerns - PSA',
                          '(Intercept)' = 'Control mean'),
             stars = TRUE ,
             gof_map = list(list('raw' = 'nobs',
                                 'clean' = 'n',
                                 'fmt' = f1)),
```

```
            escape = FALSE,
            title= 'Treatment effect estimates and response under alternative approaches to estimation
  kable_styling(latex_options = 'HOLD_position') |>
  footnote(general = paste0('\\\\footnotesize The sample is users in the evaluation stage. The response
  ),
  escape = FALSE,
  threeparttable = TRUE,
  general_title = '')
```

```
## Warning: To compile a LaTeX document with this table, the following commands must be placed in the do
##
## \usepackage{tabularray}
## \usepackage{float}
## \usepackage{graphicx}
## \usepackage{codehigh}
## \usepackage[normalem]{ulem}
## \UseTblrLibrary{booktabs}
## \UseTblrLibrary{siunitx}
## \newcommand{\tinytableTabularrayUnderline}[1]{\underline{#1}}
## \newcommand{\tinytableTabularrayStrikeout}[1]{\sout{#1}}
## \NewTableCommand{\tinytableDefineColor}[3]{\definecolor{#1}{#2}{#3}}
##
## To disable `siunitx` and prevent `modelsummary` from wrapping numeric entries in `\num{}`, call:
##
## options("modelsummary_format_numeric_latex" = "plain")
##  This warning appears once per session.
```

Table 5: Treatment effect estimates and response under alternative approaches to estimation, intention measure.

	DM	IMP	Lin	IP	GRF
Combined					
PSA - Control	0.036*	0.030+	0.021*	0.028+	0.026**
	(0.016)	(0.016)	(0.009)	(0.016)	(0.009)
Concerns - Control	0.173***	0.151***	0.122***	0.152***	0.129***
	(0.016)	(0.015)	(0.010)	(0.016)	(0.010)
Concerns - PSA	0.137***	0.121***	0.101***	0.123***	0.103***
	(0.016)	(0.015)	(0.010)	(0.016)	(0.010)
Control mean	3.157	3.155	3.181	3.166	3.177
	(0.012)	(0.011)	(0.006)	(0.011)	(0.006)
n	20,887	22,052	20,887	20,887	20,887
Kenya					
PSA - Control	0.067*	0.063*	0.035*	0.052+	0.039*
	(0.029)	(0.028)	(0.015)	(0.029)	(0.015)
Concerns - Control	0.211***	0.202***	0.154***	0.184***	0.163***
	(0.028)	(0.027)	(0.017)	(0.028)	(0.017)
Concerns - PSA	0.144***	0.139***	0.119***	0.135***	0.125***
	(0.028)	(0.027)	(0.018)	(0.028)	(0.018)
Control mean	3.163	3.159	3.194	3.178	3.189
	(0.020)	(0.020)	(0.010)	(0.020)	(0.010)
n	6,799	7,235	6,799	6,799	6,799
Nigeria					
PSA - Control	0.022	0.014	0.014	0.020	0.017
	(0.020)	(0.019)	(0.011)	(0.020)	(0.011)
Concerns - Control	0.156***	0.126***	0.105***	0.138***	0.111***
	(0.019)	(0.019)	(0.012)	(0.019)	(0.012)
Concerns - PSA	0.134***	0.112***	0.090***	0.119***	0.091***
	(0.019)	(0.019)	(0.013)	(0.019)	(0.013)
Control mean	3.154	3.153	3.175	3.160	3.171
	(0.014)	(0.014)	(0.007)	(0.014)	(0.007)
n	14,088	14,817	14,088	14,088	14,088

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

The sample is users in the evaluation stage. The response measure is the vaccine intention measure. Estimates are average treatment effects, and control mean. Estimating procedures are discussed in the text. Statistical significance is reported only for treatment effect estimates, not for control means.

```
#####################################################
### code chunk number 15: main_results_willingness
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

# Combine model objects
out_list_w <- list(
```

```
  DM = est_DM_vax_w,
  IMP = est_IMP_vax_w,
  Lin = est_Lin_vax_w,
  IP = est_IP_vax_w
)

out_list_kenya_w <- list(
  DM = est_DM_kenya_w,
  IMP = est_IMP_kenya_w,
  Lin = est_Lin_kenya_w,
  IP = est_IP_kenya_w
)

out_list_nigeria_w <- list(
  DM = est_DM_nigeria_w,
  IMP = est_IMP_nigeria_w,
  Lin = est_Lin_nigeria_w,
  IP = est_IP_nigeria_w
)

out_list_w <- modelsummary(out_list_w,
                           output = 'modelsummary_list')
out_list_kenya_w <- modelsummary(out_list_kenya_w,
                                 output = 'modelsummary_list')
out_list_nigeria_w <- modelsummary(out_list_nigeria_w,
                                   output = 'modelsummary_list')

for(x in names(out_list_w)){
  # add on Concerns - PSA coefficients
  out_list_w[[x]]$tidy <- rbind(out_list_w[[x]]$tidy |>
                                  select(term, estimate, std.error, statistic,
                                         p.value),
                                broom::tidy(get(paste0('est_', x, '_vax_alt_w'))) |>
                                  filter(term == 'treatment_groupConcerns') |>
                                  mutate(term = 'treatment_groupConcerns2') |>
                                  select(term, estimate, std.error, statistic,
                                         p.value))

  out_list_kenya_w[[x]]$tidy <- rbind(out_list_kenya_w[[x]]$tidy |>
                                        select(term, estimate, std.error, statistic,
                                               p.value),
                                      broom::tidy(get(paste0('est_', x, '_kenya_alt_w'))) |>
                                        filter(term == 'treatment_groupConcerns') |>
                                        mutate(term = 'treatment_groupConcerns2') |>
                                        select(term, estimate, std.error, statistic,
                                               p.value))

  out_list_nigeria_w[[x]]$tidy <- rbind(out_list_nigeria_w[[x]]$tidy |>
                                          select(term, estimate, std.error, statistic,
                                                 p.value),
                                        broom::tidy(get(paste0('est_', x, '_nigeria_alt_w'))) |>
                                          filter(term == 'treatment_groupConcerns') |>
                                          mutate(term = 'treatment_groupConcerns2') |>
```

```
                                        select(term, estimate, std.error, statistic,
                                               p.value))

  # drop p-values for control condition so no stars
  out_list_w[[x]]$tidy[which(out_list_w[[x]]$tidy$term == '(Intercept)'),
                       c('p.value', 'statistic')] <- NA
  out_list_kenya_w[[x]]$tidy[which(out_list_kenya_w[[x]]$tidy$term == '(Intercept)'),
                             c('p.value', 'statistic')] <- NA
  out_list_nigeria_w[[x]]$tidy[which(out_list_nigeria_w[[x]]$tidy$term == '(Intercept)'),
                               c('p.value', 'statistic')] <- NA
}

# Add GRF; different format
out_list_w[['GRF']] <- format_grf_summary(est_GRF_vax_w)
out_list_w[['GRF']]$tidy <- rbind(out_list_w[['GRF']]$tidy,
                                  format_grf_summary(est_GRF_vax_alt_w)$tidy |>
                                    filter(term == 'treatment_groupConcerns') |>
                                    mutate(term = 'treatment_groupConcerns2'))
out_list_w[['GRF']]$tidy[which(out_list_w[['GRF']]$tidy$term == '(Intercept)'),
                         c('p.value', 'statistic')] <- NA


out_list_kenya_w[['GRF']] <- format_grf_summary(est_GRF_kenya_w)
out_list_kenya_w[['GRF']]$tidy <- rbind(out_list_kenya_w[['GRF']]$tidy,
                                        format_grf_summary(est_GRF_kenya_alt_w)$tidy |>
                                          filter(term == 'treatment_groupConcerns') |>
                                          mutate(term = 'treatment_groupConcerns2'))
out_list_kenya_w[['GRF']]$tidy[which(out_list_kenya_w[['GRF']]$tidy$term == '(Intercept)'),
                               c('p.value', 'statistic')] <- NA

out_list_nigeria_w[['GRF']] <- format_grf_summary(est_GRF_nigeria_w)
out_list_nigeria_w[['GRF']]$tidy <- rbind(out_list_nigeria_w[['GRF']]$tidy,
                                          format_grf_summary(est_GRF_nigeria_alt_w)$tidy |>
                                            filter(term == 'treatment_groupConcerns') |>
                                            mutate(term = 'treatment_groupConcerns2'))
out_list_nigeria_w[['GRF']]$tidy[which(out_list_nigeria_w[['GRF']]$tidy$term == '(Intercept)'),
                                 c('p.value', 'statistic')] <- NA


modelsummary(list(Combined = out_list_w,
                  Kenya = out_list_kenya_w,
                  Nigeria = out_list_nigeria_w),
             shape= 'rbind',
             output = 'latex',
             coef_map = c('treatment_groupPSA' = 'PSA - Control',
                          'treatment_groupConcerns' = 'Concerns - Control',
                          'treatment_groupConcerns2' = 'Concerns - PSA',
                          '(Intercept)' = 'Control mean'),
             stars = TRUE ,
             gof_map = list(list('raw' = 'nobs',
                                 'clean' = 'n',
                                 'fmt' = f1)),
             escape = FALSE,
```

```
            title= 'Treatment effect estimates and response under alternative approaches to estimation
kable_styling(latex_options = c('HOLD_position')) |>
footnote(general = paste0('\\\\footnotesize The sample is users in the evaluation stage. The response
),
escape = FALSE,
threeparttable = TRUE,
general_title = '')
```

Table 6: Treatment effect estimates and response under alternative approaches to estimation, willingness measure.

	DM	IMP	Lin	IP	GRF
Combined					
PSA - Control	0.028	0.024	0.012	0.018	0.014
	(0.019)	(0.019)	(0.011)	(0.019)	(0.012)
Concerns - Control	0.156***	0.147***	0.104***	0.132***	0.111***
	(0.019)	(0.019)	(0.013)	(0.019)	(0.013)
Concerns - PSA	0.128***	0.123***	0.093***	0.113***	0.097***
	(0.019)	(0.019)	(0.013)	(0.019)	(0.013)
Control mean	2.592	2.590	2.616	2.602	2.613
	(0.014)	(0.013)	(0.008)	(0.014)	(0.006)
n	20,993	22,052	20,993	20,993	20,993
Kenya					
PSA - Control	0.071*	0.065+	0.032	0.057	0.029
	(0.035)	(0.034)	(0.021)	(0.035)	(0.021)
Concerns - Control	0.170***	0.180***	0.108***	0.142***	0.114***
	(0.035)	(0.034)	(0.023)	(0.035)	(0.023)
Concerns - PSA	0.099**	0.115***	0.076**	0.085*	0.081***
	(0.035)	(0.033)	(0.024)	(0.035)	(0.024)
Control mean	2.723	2.718	2.757	2.737	2.757
	(0.024)	(0.024)	(0.014)	(0.025)	(0.012)
n	6,840	7,235	6,840	6,840	6,840
Nigeria					
PSA - Control	0.007	0.003	0.002	0.001	0.005
	(0.023)	(0.022)	(0.014)	(0.023)	(0.014)
Concerns - Control	0.151***	0.131***	0.102***	0.130***	0.110***
	(0.023)	(0.022)	(0.015)	(0.023)	(0.015)
Concerns - PSA	0.143***	0.128***	0.100***	0.127***	0.105***
	(0.023)	(0.022)	(0.016)	(0.023)	(0.016)
Control mean	2.528	2.527	2.548	2.537	2.545
	(0.016)	(0.016)	(0.009)	(0.016)	(0.007)
n	14,153	14,817	14,153	14,153	14,153

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

The sample is users in the evaluation stage. The response measure is the vaccine willingness measure. Estimates are average treatment effects, and control mean. Estimating procedures are discussed in the text. Statistical significance is reported only for treatment effect estimates, not for control means.

```
#####################################################
### code chunk number 16: main_results_index
```

```
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

# Combine model objects
out_list <- list(
  DM = est_DM_vax,
  IMP = est_IMP_vax,
  Lin = est_Lin_vax,
  IP = est_IP_vax
)

out_list_kenya <- list(
  DM = est_DM_kenya,
  IMP = est_IMP_kenya,
  Lin = est_Lin_kenya,
  IP = est_IP_kenya
)

out_list_nigeria <- list(
  DM = est_DM_nigeria,
  IMP = est_IMP_nigeria,
  Lin = est_Lin_nigeria,
  IP = est_IP_nigeria
)

out_list <- modelsummary(out_list,
                         output = 'modelsummary_list')
out_list_kenya <- modelsummary(out_list_kenya,
                               output = 'modelsummary_list')
out_list_nigeria <- modelsummary(out_list_nigeria,
                                 output = 'modelsummary_list')

for(x in names(out_list)){
  # add on Concerns - PSA coefficients
  out_list[[x]]$tidy <- rbind(out_list[[x]]$tidy |>
                                select(term, estimate, std.error, statistic,
                                       p.value),
                              broom::tidy(get(paste0('est_', x, '_vax_alt'))) |>
                                filter(term == 'treatment_groupConcerns') |>
                                mutate(term = 'treatment_groupConcerns2') |>
                                select(term, estimate, std.error, statistic,
                                       p.value))

  out_list_kenya[[x]]$tidy <- rbind(out_list_kenya[[x]]$tidy |>
                                      select(term, estimate, std.error, statistic,
                                             p.value),
                                    broom::tidy(get(paste0('est_', x, '_kenya_alt'))) |>
                                      filter(term == 'treatment_groupConcerns') |>
                                      mutate(term = 'treatment_groupConcerns2') |>
                                      select(term, estimate, std.error, statistic,
```

```
                                          p.value))

  out_list_nigeria[[x]]$tidy <- rbind(out_list_nigeria[[x]]$tidy |>
                                        select(term, estimate, std.error, statistic,
                                               p.value),
                                      broom::tidy(get(paste0('est_', x, '_nigeria_alt'))) |>
                                        filter(term == 'treatment_groupConcerns') |>
                                        mutate(term = 'treatment_groupConcerns2') |>
                                        select(term, estimate, std.error, statistic,
                                               p.value))

  # drop p-values for control condition so no stars
  out_list[[x]]$tidy[which(out_list[[x]]$tidy$term == '(Intercept)'),
                     c('p.value', 'statistic')] <- NA
  out_list_kenya[[x]]$tidy[which(out_list_kenya[[x]]$tidy$term == '(Intercept)'),
                           c('p.value', 'statistic')] <- NA
  out_list_nigeria[[x]]$tidy[which(out_list_nigeria[[x]]$tidy$term == '(Intercept)'),
                             c('p.value', 'statistic')] <- NA
}

# Add GRF; different format
out_list[['GRF']] <- format_grf_summary(est_GRF_vax)
out_list[['GRF']]$tidy <- rbind(out_list[['GRF']]$tidy,
                                format_grf_summary(est_GRF_vax_alt)$tidy |>
                                  filter(term == 'treatment_groupConcerns') |>
                                  mutate(term = 'treatment_groupConcerns2'))
out_list[['GRF']]$tidy[which(out_list[['GRF']]$tidy$term == '(Intercept)'),
                       c('p.value', 'statistic')] <- NA


out_list_kenya[['GRF']] <- format_grf_summary(est_GRF_kenya)
out_list_kenya[['GRF']]$tidy <- rbind(out_list_kenya[['GRF']]$tidy,
                                      format_grf_summary(est_GRF_kenya_alt)$tidy |>
                                        filter(term == 'treatment_groupConcerns') |>
                                        mutate(term = 'treatment_groupConcerns2'))
out_list_kenya[['GRF']]$tidy[which(out_list_kenya[['GRF']]$tidy$term == '(Intercept)'),
                             c('p.value', 'statistic')] <- NA

out_list_nigeria[['GRF']] <- format_grf_summary(est_GRF_nigeria)
out_list_nigeria[['GRF']]$tidy <- rbind(out_list_nigeria[['GRF']]$tidy,
                                        format_grf_summary(est_GRF_nigeria_alt)$tidy |>
                                          filter(term == 'treatment_groupConcerns') |>
                                          mutate(term = 'treatment_groupConcerns2'))
out_list_nigeria[['GRF']]$tidy[which(out_list_nigeria[['GRF']]$tidy$term == '(Intercept)'),
                               c('p.value', 'statistic')] <- NA


modelsummary(list(Combined = out_list,
                  Kenya = out_list_kenya,
                  Nigeria = out_list_nigeria),
             shape= 'rbind',
             output = 'latex',
             coef_map = c('treatment_groupPSA' = 'PSA - Control',
```

```
                        'treatment_groupConcerns' = 'Concerns - Control',
                        'treatment_groupConcerns2' = 'Concerns - PSA',
                        '(Intercept)' = 'Control mean'),
            stars = TRUE ,
            gof_map = list(list('raw' = 'nobs',
                                'clean' = 'n',
                                'fmt' = f1)),
            escape = FALSE,
            title= 'Treatment effect estimates and response under alternative approaches to estimation
kable_styling(latex_options = c('HOLD_position')) |>
footnote(general = paste0('\\\\footnotesize The sample is users in the evaluation stage. The response
),
escape = FALSE,
threeparttable = TRUE,
general_title = '')
```

Table 7: Treatment effect estimates and response under alternative approaches to estimation, combined response function.

	DM	IMP	Lin	IP	GRF
Combined					
PSA - Control	0.065*	0.053+	0.033*	0.049	0.040**
	(0.030)	(0.030)	(0.015)	(0.030)	(0.015)
Concerns - Control	0.324***	0.289***	0.221***	0.280***	0.235***
	(0.030)	(0.029)	(0.017)	(0.030)	(0.017)
Concerns - PSA	0.259***	0.236***	0.189***	0.232***	0.193***
	(0.030)	(0.029)	(0.018)	(0.030)	(0.018)
Control mean	-0.123	-0.115	-0.075	-0.104	-0.082
	(0.022)	(0.021)	(0.010)	(0.021)	(0.011)
n	20,825	22,052	20,825	20,825	20,825
Kenya					
PSA - Control	0.133*	0.124*	0.065*	0.108*	0.074**
	(0.055)	(0.054)	(0.026)	(0.055)	(0.027)
Concerns - Control	0.375***	0.372***	0.257***	0.324***	0.275***
	(0.053)	(0.052)	(0.030)	(0.054)	(0.031)
Concerns - PSA	0.242***	0.248***	0.192***	0.218***	0.200***
	(0.054)	(0.052)	(0.032)	(0.054)	(0.033)
Control mean	0.000	0.003	0.062	0.025	0.056
	(0.039)	(0.038)	(0.017)	(0.039)	(0.020)
n	6,784	7,235	6,784	6,784	6,784
Nigeria					
PSA - Control	0.032	0.017	0.017	0.029	0.022
	(0.036)	(0.035)	(0.018)	(0.036)	(0.018)
Concerns - Control	0.300***	0.248***	0.201***	0.266***	0.214***
	(0.036)	(0.035)	(0.021)	(0.036)	(0.021)
Concerns - PSA	0.268***	0.231***	0.184***	0.239***	0.194***
	(0.035)	(0.034)	(0.022)	(0.036)	(0.022)
Control mean	-0.182	-0.172	-0.141	-0.169	-0.146
	(0.026)	(0.025)	(0.012)	(0.026)	(0.013)
n	14,041	14,817	14,041	14,041	14,041

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

The sample is users in the evaluation stage. The response measure is the combined index. Estimates are average treatment effects, and control mean. Estimating procedures are discussed in the text. Statistical significance is reported only for treatment effect estimates, not for control means.

```
#####################################################
### code chunk number 17: secondary_estimation
#####################################################
#| eval = TRUE,
#| cache = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

## outcomes of interest ##
```

```r
## 1. sharing: (0,.5,1) - proportion of stimuli wanted to share (messenger and timeline)
# true: "share_true_scaled"
# false: "share_false_scaled"
## 2. Information seeking
# "pre_register_imputed"
df_eval$pre_register <- df_eval$pre_register_imputed
## 3. encouraging others - post a WHO pro-vaccine msg on their timeline
# "share_post_numeric"
## 4. vaccination status (from follow up survey)
# "vaxed_f_numeric" / "vaxed_f_binary"

# Function for each model:

run_DM <- function(outcome, data, data_alt) {
  est_reg <- estimatr::lm_robust(as.formula(paste0(outcome, " ~ treatment_group")), data = data)
  est_reg_alt <- estimatr::lm_robust(as.formula(paste0(outcome, " ~ treatment_group")), data = data_alt)
  return(list(est_reg, est_reg_alt))
}

run_GRF <- function(outcome, data, data_alt, covariates) {
  if(outcome == 'vaxed_f_binary') {
    covariates <- c(covariates, 'response_imputed')
  }

  est_reg <- mcf_estimate(data, outcome, covariates = covariates)
  est_reg_alt <- mcf_estimate(data_alt, outcome, covariates = covariates)
  return(list(est_reg, est_reg_alt))
}

run_IP <- function(outcome, data, data_alt, covariates) {
  if(outcome == 'vaxed_f_binary') {
    covariates <- c(covariates, 'response_imputed')
  }
  est_reg <- IP_estimate(data, outcome, covariates = covariates)[['ATE']]
  est_reg_alt <- IP_estimate(data_alt, outcome, covariates = covariates)[['ATE']]
  return(list(est_reg, est_reg_alt))
}

run_Lin <- function(outcome, data, data_alt, covariates_lin) {
  if(outcome == 'vaxed_f_binary') {
    covariates_lin <- as.formula(paste('~' , paste0(
      c('willingness', 'get_vaccinated'),
      collapse = ' + ')))
  }

  est_reg <- estimatr::lm_lin(as.formula(paste0(outcome, " ~ treatment_group")),
                              data = data, covariates = covariates_lin)
  est_reg_alt <- estimatr::lm_lin(as.formula(paste0(outcome, " ~ treatment_group")),
                                  data = data_alt, covariates = covariates_lin)
  return(list(est_reg, est_reg_alt))
}
```

```r
# Main function:
# Attrition / missing data:
# - For information seeking and encouragement, we impute zeros, i.e., that
# respondents did not click through for more information or share posts with
# peers, which is in line with their actual behavior.
# - For sharing behavior, we report both unadjusted, and re-weighted doubly
# robust estimates, where the weighting accounts for both treatment propensity
# and missingness propensity, under the assumption that missingness is ignorable
# conditioning on measured covariates and treatment.
# - In the follow-up survey where we elicit vaccination status, we report both
# unadjusted, and re-weighted doubly robust estimates, where the weighting
# accounts for both treatment propensity and missingness propensity, under the
# assumption that missingness is ignorable conditioning on measured covariates,
# treatment, and **primary response**.


run_all_models <- function(data, data_alt, covariate_list, covariate_lin) {
  outcomes <- c("share_true_scaled", # sharing behavior
                "share_false_scaled", # sharing behavior
                "pre_register_imputed", # information seeking
                "share_post_numeric_imputed", # encouragement
                "vaxed_f_binary" # vaccination status
  )
  results <- list()

  for (outcome in outcomes) {
    results[[outcome]] <- list(
      DM = run_DM(outcome, data, data_alt),
      GRF = run_GRF(outcome, data, data_alt, covariate_list),
      IP = run_IP(outcome, data, data_alt, covariate_list),
      Lin = run_Lin(outcome, data, data_alt, covariate_lin)
    )

  }
  return(results)
}

# run function for all 3 datasets:

results <- run_all_models(df_eval, df_eval_alt, covariate_list, covariate_lin)
```

```
## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## Control: 0.017 PSA: 0.082 Concerns: 0.209
## and the maximum is
## Control: 0.481 PSA: 0.481 Concerns: 0.867.
```

```
## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## PSA: 0.106 Control: 0.009 Concerns: 0.201
## and the maximum is
## PSA: 0.496 Control: 0.476 Concerns: 0.864.
```

```
## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## Control: 0.016 PSA: 0.096 Concerns: 0.21
## and the maximum is
## Control: 0.468 PSA: 0.505 Concerns: 0.864.
```

```
## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## PSA: 0.096 Control: 0.008 Concerns: 0.219
## and the maximum is
## PSA: 0.462 Control: 0.456 Concerns: 0.89.

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## Control: 0.021 PSA: 0.022 Concerns: 0.28
## and the maximum is
## Control: 0.49 PSA: 0.532 Concerns: 0.9.

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## PSA: 0.024 Control: 0.022 Concerns: 0.223
## and the maximum is
## PSA: 0.561 Control: 0.361 Concerns: 0.913.
results_kenya <- run_all_models(df_kenya, df_kenya_alt, covariate_list, covariate_lin)

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## Control: 0.017 PSA: 0.104 Concerns: 0.221
## and the maximum is
## Control: 0.444 PSA: 0.464 Concerns: 0.854.

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## PSA: 0.084 Control: 0.008 Concerns: 0.194
## and the maximum is
## PSA: 0.457 Control: 0.45 Concerns: 0.895.

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## Control: 0.014 PSA: 0.095 Concerns: 0.225
## and the maximum is
## Control: 0.455 PSA: 0.458 Concerns: 0.874.

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## PSA: 0.093 Control: 0.007 Concerns: 0.216
## and the maximum is
## PSA: 0.457 Control: 0.446 Concerns: 0.89.
results_nigeria <- run_all_models(df_nigeria, df_nigeria_alt, covariate_list, covariate_lin)

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## Control: 0.024 PSA: 0.091 Concerns: 0.22
## and the maximum is
## Control: 0.487 PSA: 0.441 Concerns: 0.873.

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## PSA: 0.084 Control: 0.016 Concerns: 0.221
## and the maximum is
## PSA: 0.457 Control: 0.477 Concerns: 0.889.

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## Control: 0.019 PSA: 0.085 Concerns: 0.21
## and the maximum is
## Control: 0.488 PSA: 0.451 Concerns: 0.871.

## Warning in get_scores.multi_arm_causal_forest(forest, subset = subset, debiasing.weights = debiasing
## PSA: 0.086 Control: 0.014 Concerns: 0.198
## and the maximum is
## PSA: 0.476 Control: 0.491 Concerns: 0.874.
```

```r
# code to extract coefficients, se's and n's

extract_coefficients_and_se <- function(results) {
  coefficients <- list()

  # Specifying the coefficient mapping
  coef_map <- list(
    'treatment_groupPSA' = 'PSA - Control',
    'treatment_groupConcerns' = c('Concerns - Control', 'Concerns - PSA'),
    '(Intercept)' = 'Control mean'
  )

  for (outcome in names(results)) {
    coefficients[[outcome]] <- list()

    for (model_type in names(results[[outcome]])) {
      # Create new lists to store the specific coefficients and standard errors
      specific_coefs <- list()
      specific_se <- list()
      specific_nobs <- list()

      if (model_type != "GRF") {
        # Extracting coefficients and standard errors for regular data
        est_reg <- results[[outcome]][[model_type]][[1]]
        est_reg_alt <- results[[outcome]][[model_type]][[2]]

        # Extract nobs
        specific_nobs$main <- est_reg$nobs
        specific_nobs$alt <- est_reg_alt$nobs

        # Loop through the coef_map and extract relevant coefficients and standard errors
        for (coef_name in names(coef_map)) {
          if (coef_name %in% names(est_reg$coefficients)) {
            specific_coefs[[coef_map[[coef_name]][1]]] <- est_reg$coefficients[[coef_name]]
            specific_se[[coef_map[[coef_name]][1]]] <- est_reg$std.error[[coef_name]]
          }

          if (length(coef_map[[coef_name]]) > 1 && coef_name %in% names(est_reg_alt$coefficients)) {
            specific_coefs[[coef_map[[coef_name]][2]]] <- est_reg_alt$coefficients[[coef_name]]
            specific_se[[coef_map[[coef_name]][2]]] <- est_reg_alt$std.error[[coef_name]]
          }
        }
      } else {
        # For GRF, use the format_grf_summary function
        grf_summary <- format_grf_summary(results[[outcome]][[model_type]][[1]])$tidy
        grf_summary_alt <- format_grf_summary(results[[outcome]][[model_type]][[2]])$tidy

        specific_nobs$main <- results[[outcome]][[model_type]][[1]]$nobs[1]
        specific_nobs$alt <- results[[outcome]][[model_type]][[2]]$nobs[1]

        # Extracting coefficients and standard errors for GRF. Adjust if the structure is different.
        for (coef_name in names(coef_map)) {
          if (coef_name %in% grf_summary$term) {
```

```
            specific_coefs[[coef_map[[coef_name]][1]]] <- grf_summary[which(grf_summary$term == coef_nam
            specific_se[[coef_map[[coef_name]][1]]] <- grf_summary[which(grf_summary$term == coef_name)
          }

          if (length(coef_map[[coef_name]]) > 1 && coef_name %in% grf_summary_alt$term) {
            specific_coefs[[coef_map[[coef_name]][2]]] <- grf_summary_alt[which(grf_summary_alt$term ==
            specific_se[[coef_map[[coef_name]][2]]] <- grf_summary_alt[which(grf_summary_alt$term == coe
          }
        }
      }

      coefficients[[outcome]][[model_type]] <- list(
        coef = specific_coefs,
        se = specific_se,
        nobs = specific_nobs
      )
    }
  }

  return(coefficients)
}



#####################################################
### code chunk number 18: extract_results
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
coefficients <- extract_coefficients_and_se(results)
coefficients_kenya <- extract_coefficients_and_se(results_kenya)
coefficients_nigeria <- extract_coefficients_and_se(results_nigeria)


#####################################################
### code chunk number 19: secondary_results_all_new
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

generate_kable_table_per_outcome <- function(outcome_coefficients) {
  # Dataset to label mapping
  dataset_map <- list(
    'coefficients' = '\\textit{Combined}', # Italicize the sample name
    'coefficients_kenya' = '\\textit{Kenya}',  # Italicize the sample name
    'coefficients_nigeria' = '\\textit{Nigeria}'  # Italicize the sample name
  )

  outcome_map <- list(
    'true' = 'Sharing True Post',
    'false' = 'Sharing False Post',
```

```
  'register' = 'Informating Seeking',
  'encourage' = 'Encourage Others',
  'vaxed' = 'Vaccination status (follow up)'
)

rows_list <- list()

for (dataset in names(outcome_coefficients)) {
  coefficients <- outcome_coefficients[[dataset]]

  dataset_rows <- data.frame(
    Coefficient = c('PSA - Control', '', 'Concerns - Control', '', 'Concerns - PSA', '', 'Control mea
    DM = NA,
    Lin = NA,
    IP = NA,
    GRF = NA,
    stringsAsFactors = FALSE
  )

  for (coef_name in c('PSA - Control', 'Concerns - Control', 'Concerns - PSA', 'Control mean')) {
    for (model_type in c("DM", "Lin", "IP", "GRF")) {
      coef_val <- coefficients[[model_type]]$coef[[coef_name]]
      se_val <- coefficients[[model_type]]$se[[coef_name]]
      p_val <- 2 * (1 - pnorm(abs(coef_val / se_val)))  # Two-tailed p-value

      # Determine the significance symbol
      sig_symbol <- ""
      if (coef_name != 'Control mean') {
        if (p_val < 0.001) sig_symbol <- "***"
        else if (p_val < 0.01) sig_symbol <- "**"
        else if (p_val < 0.05) sig_symbol <- "*"
        else if (p_val < 0.1) sig_symbol <- "+"
      }

      coef_row_index <- which(dataset_rows$Coefficient == coef_name)
      se_row_index <- coef_row_index + 1  # SE row is right after coef row

      dataset_rows[coef_row_index, model_type] <- sprintf("%.3f%s", coef_val, sig_symbol)
      dataset_rows[se_row_index, model_type] <- sprintf("(%.3f)", se_val)
    }
  }

  # Add the nobs to the dataset rows and format it with commas
  for (model_type in c("DM", "Lin", "IP", "GRF")) {
    dataset_rows[dataset_rows$Coefficient == 'n', model_type] <- formatC(coefficients[[model_type]]$n
  }

  # Add dataset label to the top of each section
  dataset_label_row <- data.frame(Coefficient = dataset_map[[dataset]],
                                  DM = NA,
                                  Lin = NA, IP = NA, GRF = NA,
                                  stringsAsFactors = FALSE)
  rows_list[[dataset]] <- rbind(dataset_label_row, dataset_rows)
```

```
  }

  # Combine all datasets into one data frame
  full_table <- do.call(rbind, rows_list)

  # Define the caption and footnote for the table based on the outcome
  input_name <- deparse(substitute(outcome_coefficients))
  outcome_name <- sub("_.*", "", input_name)

  caption_text <- sprintf("Treatment effect estimates and response under alternative approaches to esti

  footnote_text <- "\\multicolumn{5}{l}{\\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p

  # Use kable to format the table
  kable_output <- kable(full_table,
                        format = "latex",
                        booktabs = TRUE,
                        linesep = "",
                        align = c("l", "c", "c", "c", "c"),
                        row.names = FALSE,
                        col.names = c("", "DM",
                                      "Lin", "IP", "GRF"),
                        caption = caption_text,
                        escape = FALSE) |>
    kable_styling(latex_options = c('HOLD_position')) |>
    footnote(general = paste0('\\\\footnotesize The sample is users in the evaluation stage. The respons
    ),
    escape = FALSE,
    threeparttable = TRUE,
    general_title = '') |>
    # Add lines after the dataset names
    row_spec(1, extra_latex_after = "\\midrule") |>
    row_spec(11, extra_latex_after = "\\midrule") |>
    row_spec(21, extra_latex_after = "\\midrule") |>
    row_spec(30, extra_latex_after = footnote_text)

  return(kable_output)
}



##################################################
### code chunk number 20: secondary_results_true
##################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

true_coefficients <- list(
  coefficients = coefficients[["share_true_scaled"]],
  coefficients_kenya = coefficients_kenya[["share_true_scaled"]],
  coefficients_nigeria = coefficients_nigeria[["share_true_scaled"]]
```

```
)

generate_kable_table_per_outcome(true_coefficients)
```

Table 8: Treatment effect estimates and response under alternative approaches to estimation, Sharing True Post.

	DM	Lin	IP	GRF
Combined				
PSA - Control	0.010	0.010	0.010	0.011
	(0.008)	(0.007)	(0.008)	(0.007)
Concerns - Control	-0.001	-0.010	-0.005	-0.008
	(0.008)	(0.007)	(0.008)	(0.007)
Concerns - PSA	-0.011	-0.021**	-0.015+	-0.021**
	(0.008)	(0.007)	(0.008)	(0.007)
Control mean	0.565	0.568	0.566	0.567
	(0.005)	(0.005)	(0.005)	(0.001)
n	19,886	19,886	19,886	19,886
Kenya				
PSA - Control	0.021	0.019	0.020	0.020
	(0.013)	(0.013)	(0.014)	(0.013)
Concerns - Control	0.004	-0.008	-0.000	-0.003
	(0.014)	(0.013)	(0.014)	(0.013)
Concerns - PSA	-0.018	-0.027*	-0.021	-0.025+
	(0.014)	(0.013)	(0.014)	(0.013)
Control mean	0.558	0.563	0.560	0.562
	(0.010)	(0.009)	(0.010)	(0.002)
n	6,441	6,441	6,441	6,441
Nigeria				
PSA - Control	0.005	0.007	0.004	0.006
	(0.009)	(0.008)	(0.009)	(0.008)
Concerns - Control	-0.003	-0.013	-0.008	-0.013
	(0.009)	(0.009)	(0.009)	(0.009)
Concerns - PSA	-0.007	-0.020*	-0.012	-0.018*
	(0.009)	(0.009)	(0.009)	(0.009)
Control mean	0.568	0.571	0.570	0.571
	(0.007)	(0.006)	(0.007)	(0.002)
n	13,445	13,445	13,445	13,445
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001				

The sample is users in the evaluation stage. The response measure is Sharing True Post. Estimates are average treatment effects, and control mean. Estimating procedures are discussed in the text. Statistical significance is reported only for treatment effect estimates, and differences, not for baseline control means.

```
#####################################################
### code chunk number 21: secondary_results_false
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
```

```r
#| results='asis'

false_coefficients <- list(
  coefficients = coefficients[["share_false_scaled"]],
  coefficients_kenya = coefficients_kenya[["share_false_scaled"]],
  coefficients_nigeria = coefficients_nigeria[["share_false_scaled"]]
)

print(generate_kable_table_per_outcome(false_coefficients))
```

```
## \begin{table}[H]
## \centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lcccc}
## \toprule
##  & DM & Lin & IP & GRF\\
## \midrule
## \textit{Combined} &  &  &  & \\
## \midrule
## PSA - Control & -0.006 & -0.005 & -0.005 & -0.002\\
##  & (0.007) & (0.007) & (0.007) & (0.006)\\
## Concerns - Control & -0.003 & -0.005 & -0.005 & -0.004\\
##  & (0.007) & (0.007) & (0.007) & \vphantom{1} (0.007)\\
## Concerns - PSA & 0.004 & -0.001 & 0.000 & -0.003\\
##  & (0.007) & (0.007) & (0.007) & (0.007)\\
## Control mean & 0.308 & 0.308 & 0.308 & 0.308\\
##  & (0.005) & (0.005) & (0.005) & (0.001)\\
## n & 20,124 & 20,124 & 20,124 & 20,124\\
## \textit{Kenya} &  &  &  & \\
## \midrule
## PSA - Control & -0.007 & -0.006 & -0.005 & -0.002\\
##  & (0.011) & (0.011) & (0.011) & \vphantom{2} (0.011)\\
## Concerns - Control & 0.010 & 0.009 & 0.009 & 0.010\\
##  & (0.011) & (0.011) & (0.011) & \vphantom{1} (0.011)\\
## Concerns - PSA & 0.016 & 0.015 & 0.014 & 0.013\\
##  & (0.011) & (0.011) & (0.011) & (0.011)\\
## Control mean & 0.224 & 0.224 & 0.224 & 0.223\\
##  & (0.008) & (0.008) & (0.008) & (0.001)\\
## n & 6,482 & 6,482 & 6,482 & 6,482\\
## \textit{Nigeria} &  &  &  & \\
## \midrule
## PSA - Control & -0.006 & -0.002 & -0.004 & -0.002\\
##  & (0.009) & (0.008) & (0.009) & \vphantom{2} (0.008)\\
## Concerns - Control & -0.009 & -0.012 & -0.011 & -0.011\\
##  & (0.009) & (0.008) & (0.009) & \vphantom{1} (0.008)\\
## Concerns - PSA & -0.003 & -0.010 & -0.007 & -0.009\\
##  & (0.009) & (0.008) & (0.009) & (0.008)\\
## Control mean & 0.348 & 0.348 & 0.348 & 0.347\\
##  & (0.006) & (0.006) & (0.006) & (0.002)\\
## n & 13,642 & 13,642 & 13,642 & 13,642\\
## \multicolumn{5}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \bottomrule
```

```
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is Sharing Fals
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
#####################################################
### code chunk number 22: secondary_results_register
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

register_coefficients <- list(
  coefficients = coefficients[["pre_register_imputed"]],
  coefficients_kenya = coefficients_kenya[["pre_register_imputed"]],
  coefficients_nigeria = coefficients_nigeria[["pre_register_imputed"]]
)

print(generate_kable_table_per_outcome(register_coefficients))
```

```
## \begin{table}[H]
## \centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approac
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lcccc}
## \toprule
##  & DM & Lin & IP & GRF\\
## \midrule
## \textit{Combined} &  &  &  & \\
## \midrule
## PSA - Control & 0.001 & 0.001 & -0.009*** & 0.001\\
##  & (0.003) & (0.003) & (0.002) & \vphantom{1} (0.001)\\
## Concerns - Control & -0.000 & -0.001 & -0.018*** & 0.001\\
##  & (0.003) & (0.003) & (0.002) & (0.001)\\
## Concerns - PSA & -0.002 & -0.002 & -0.008*** & 0.000\\
##  & (0.003) & (0.003) & (0.001) & (0.001)\\
## Control mean & 0.029 & 0.030 & 0.022 & 0.027\\
##  & (0.002) & (0.002) & (0.002) & \vphantom{1} (0.000)\\
## n & 22,052 & 22,052 & 22,052 & 22,052\\
## \textit{Kenya} &  &  &  & \\
## \midrule
## PSA - Control & -0.006 & -0.006 & -0.011** & -0.003\\
##  & (0.005) & (0.005) & (0.004) & (0.002)\\
## Concerns - Control & -0.009* & -0.010* & -0.021*** & -0.004\\
##  & (0.005) & (0.005) & (0.003) & (0.002)\\
## Concerns - PSA & -0.003 & -0.004 & -0.011*** & -0.001\\
##  & (0.004) & (0.004) & (0.002) & \vphantom{1} (0.002)\\
## Control mean & 0.031 & 0.032 & 0.027 & 0.027\\
##  & (0.004) & (0.004) & (0.003) & (0.000)\\
## n & 7,235 & 7,235 & 7,235 & 7,235\\
## \textit{Nigeria} &  &  &  & \\
```

```
## \midrule
## PSA - Control & 0.005 & 0.004 & -0.001 & 0.003\\
##  & (0.003) & (0.003) & (0.003) & (0.002)\\
## Concerns - Control & 0.004 & 0.003 & -0.013*** & 0.004+\\
##  & (0.003) & (0.003) & (0.002) & (0.002)\\
## Concerns - PSA & -0.001 & -0.001 & -0.013*** & 0.001\\
##  & (0.004) & (0.004) & (0.002) & (0.002)\\
## Control mean & 0.028 & 0.029 & 0.025 & 0.025\\
##  & (0.002) & (0.002) & (0.002) & (0.000)\\
## n & 14,817 & 14,817 & 14,817 & 14,817\\
## \multicolumn{5}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \bottomrule
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is Informating
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
#####################################################
### code chunk number 23: secondary_results_encourage
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

encourage_coefficients <- list(
  coefficients = coefficients[["share_post_numeric_imputed"]],
  coefficients_kenya = coefficients_kenya[["share_post_numeric_imputed"]],
  coefficients_nigeria = coefficients_nigeria[["share_post_numeric_imputed"]]
)

print(generate_kable_table_per_outcome(encourage_coefficients))
```

```
## \begin{table}[H]
## \centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lcccc}
## \toprule
##  & DM & Lin & IP & GRF\\
## \midrule
## \textit{Combined} &  &  &  & \\
## \midrule
## PSA - Control & -0.006 & -0.004 & -0.167*** & 0.001\\
##  & (0.008) & (0.008) & (0.020) & (0.002)\\
## Concerns - Control & -0.041*** & -0.046*** & -0.300*** & -0.003\\
##  & (0.008) & (0.008) & (0.016) & (0.002)\\
## Concerns - PSA & -0.035*** & -0.042*** & -0.148*** & -0.004*\\
##  & (0.008) & (0.008) & (0.012) & (0.002)\\
## Control mean & 0.510 & 0.511 & 0.358 & 0.491\\
##  & (0.006) & (0.005) & (0.016) & (0.001)\\
## n & 22,052 & 22,052 & 22,052 & 22,052\\
```

```
## \textit{Kenya} &  &  &  & \\
## \midrule
## PSA - Control & -0.013 & -0.017 & -0.114*** & -0.005\\
##  & (0.014) & (0.014) & (0.023) & (0.007)\\
## Concerns - Control & -0.046** & -0.058*** & -0.330*** & -0.007\\
##  & (0.014) & (0.014) & (0.018) & (0.007)\\
## Concerns - PSA & -0.033* & -0.042** & -0.214*** & -0.005\\
##  & (0.014) & (0.014) & (0.017) & (0.006)\\
## Control mean & 0.514 & 0.519 & 0.450 & 0.491\\
##  & (0.010) & (0.010) & (0.017) & (0.002)\\
## n & 7,235 & 7,235 & 7,235 & 7,235\\
## \textit{Nigeria} &  &  &  & \\
## \midrule
## PSA - Control & -0.003 & 0.002 & -0.082*** & 0.004\\
##  & (0.010) & (0.009) & (0.018) & (0.005)\\
## Concerns - Control & -0.038*** & -0.042*** & -0.273*** & -0.008\\
##  & (0.010) & (0.009) & (0.014) & (0.005)\\
## Concerns - PSA & -0.035*** & -0.044*** & -0.200*** & -0.011*\\
##  & (0.010) & (0.009) & (0.015) & (0.005)\\
## Control mean & 0.508 & 0.509 & 0.444 & 0.490\\
##  & (0.007) & (0.007) & (0.012) & (0.002)\\
## n & 14,817 & 14,817 & 14,817 & 14,817\\
## \multicolumn{5}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \bottomrule
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is Encourage O
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
#####################################################
### code chunk number 24: secondary_results_vaxed
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

vaxed_coefficients <- list(
  coefficients = coefficients[["vaxed_f_binary"]],
  coefficients_kenya = coefficients_kenya[["vaxed_f_binary"]],
  coefficients_nigeria = coefficients_nigeria[["vaxed_f_binary"]]
)

print(generate_kable_table_per_outcome(vaxed_coefficients))
```

```
## \begin{table}[H]
## \centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lcccc}
## \toprule
##  & DM & Lin & IP & GRF\\
```

```
## \midrule
## \textit{Combined} &  &  &  & \\
## \midrule
## PSA - Control & -0.005 & -0.014 & 0.201 & 2.074\\
##  & (0.045) & (0.045) & (0.163) & (1.771)\\
## Concerns - Control & -0.019 & -0.025 & 0.127 & 0.359\\
##  & (0.045) & (0.045) & (0.154) & (0.249)\\
## Concerns - PSA & -0.015 & -0.011 & -0.083 & -2.022\\
##  & (0.046) & (0.045) & (0.227) & (1.979)\\
## Control mean & 0.272 & 0.279 & 0.241 & 0.270\\
##  & (0.031) & (0.032) & (0.043) & (0.004)\\
## n & 563 & 560 & 563 & 563\\
## \textit{Kenya} &  &  &  & \\
## \midrule
## PSA - Control & -0.024 & -0.039 & 0.026 & 0.032\\
##  & (0.080) & (0.080) & (0.092) & (0.089)\\
## Concerns - Control & 0.013 & -0.014 & 0.026 & 0.060\\
##  & (0.086) & (0.081) & (0.204) & (0.187)\\
## Concerns - PSA & 0.037 & 0.025 & -0.011 & 0.052\\
##  & (0.085) & (0.080) & (0.204) & (0.218)\\
## Control mean & 0.338 & 0.356 & 0.284 & 0.308\\
##  & (0.058) & (0.058) & (0.073) & (0.010)\\
## n & 195 & 193 & 195 & 195\\
## \textit{Nigeria} &  &  &  & \\
## \midrule
## PSA - Control & 0.000 & -0.003 & 0.071 & 0.112\\
##  & (0.055) & (0.055) & (0.093) & (0.124)\\
## Concerns - Control & -0.032 & -0.033 & 0.095 & 0.128\\
##  & (0.052) & (0.053) & (0.141) & (0.134)\\
## Concerns - PSA & -0.032 & -0.029 & 0.041 & 0.063\\
##  & (0.055) & (0.055) & (0.175) & (0.178)\\
## Control mean & 0.239 & 0.242 & 0.241 & 0.222\\
##  & (0.037) & (0.037) & (0.037) & (0.002)\\
## n & 368 & 367 & 368 & 368\\
## \multicolumn{5}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \bottomrule
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is Vaccination
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
#####################################################
### code chunk number 25: predict_type
#####################################################
#| eval = TRUE,
#| echo=FALSE,
#| warning=FALSE,
#| message=FALSE,
#| strip.white=TRUE,
#| results='asis'

summary_variables <- c(
```

```
  #demographics
  'age', 'is_female', 'religion_christian', 'religiosity', 'party_aligned',
  #SES
  'is_urban', 'completed_secondary_school', 'hhold20', 'assets_index', 'has_job',
  #healthcare access, knowledge, attitudes
  'health_access',
  'cov_know',
  'know_wheretoget',
  'adult_vax',
  'getvax_easy',
  'trust_index')

# create groups based on two pre-treatment measures of vaccine willingness/intention (1=Hesitant, 2/3 =
df_eval <- df_eval |>
  mutate(willingness_group =
           factor(case_when(willingness > 2 ~ willingness-1,
                            willingness <= 2 ~ willingness,
                            TRUE ~ 0),
                  levels = c(0,1,2,3),
                  labels = c('Missing', 'Hesitant','Undecided','Eager')),
         willingness_group_post =
           factor(case_when(willingness_2 > 2 ~ willingness_2-1,
                            willingness_2 <= 2 ~ willingness_2,
                            TRUE ~ 0),
                  levels = c(0,1,2,3),
                  labels = c('Missing', 'Hesitant','Undecided','Eager')),
         get_vaccinated_group =
           factor(case_when(get_vaccinated > 2 ~ get_vaccinated-1,
                            get_vaccinated <= 2 ~ get_vaccinated,
                            TRUE ~ 0),
                  levels = c(0,1,2,3),
                  labels = c('Missing', 'Hesitant','Undecided','Eager')),
         get_vaccinated_group_post =
           factor(case_when(get_vaccinated_2 > 2 ~ get_vaccinated_2-1,
                            get_vaccinated_2 <= 2 ~ get_vaccinated_2,
                            TRUE ~ 0),
                  levels = c(0,1,2,3),
                  labels = c('Missing', 'Hesitant','Undecided','Eager')))

df_willingness <- df_eval |>
  mutate(is_female = 1*(is_male==0),
         hhold20 = case_when(hhold<20 ~hhold,
                             TRUE ~NA)) |>
  drop_na(willingness_group) |>
  group_by(willingness_group) |>
  summarise(across(all_of(summary_variables),
                   .fns = list(mean = ~ sprintf('%.3f', mean(., na.rm = TRUE)),
                               se = ~ sprintf('(%.3f)',
                                              sd(., na.rm = TRUE)/sqrt(sum(!is.na(.))))),
                   .names = '{.fn}_{.col}'
  )) |> t()
```

```r
colnames(df_willingness) <- df_willingness['willingness_group',]
row.names(df_willingness)[grep('se_', row.names(df_willingness))] <- ''

label_rows <- c('mean_age' = 'Age',
                'mean_is_female' = '% Non-male',
                'mean_health_access' = 'Hours to nearest health facility',
                'mean_party_aligned' = '% Party aligned',
                'mean_cov_know' = 'COVID knowledge index (-5:5)',
                'mean_religion_christian' = '% Christian',
                'mean_religiosity' = 'Religiosity (1:6)',
                'mean_completed_secondary_school' = '% Completed secondary school',
                'mean_is_urban' = '% Urban',
                'mean_hhold20' = 'Household size',
                'mean_assets_index' = 'Assets index (-5:5)',
                'mean_has_job' = '% Employed',
                'mean_trust_index' = 'Trust index (-14:14)',
                'mean_adult_vax' = '% Any prior vaccination',
                'mean_know_wheretoget' = '% Know where to get COVID vaccine',
                'mean_getvax_easy' = 'Ease of getting COVID vaccine (-2:2)')

rownames(df_willingness)[match(names(label_rows),
                               rownames(df_willingness))] <- label_rows

kbl(df_willingness[-1,],
    format = 'latex',
    caption= 'Composition of baseline vaccine attitude types.',
    align = 'c', booktabs = TRUE,
    toprule = '\\vspace{-3em} \\\\ \\toprule') |>
  kable_styling(latex_options = c('HOLD_position')) |>
  pack_rows('Demographics', 2, 3) |>
  pack_rows(NULL, 4, 5, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 6, 7, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 8, 9, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 10, 11, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 12, 13, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows('Socioeconomic status', 14, 15) |>
  pack_rows(NULL, 16, 17, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 18, 19, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 20, 21, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 22, 23, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows('Healthcare access/knowledge/attitudes', 24, 25) |>
  pack_rows(NULL, 26, 27, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
            escape = FALSE, colnum = 0) |>
  pack_rows(NULL, 28, 29, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
```

```
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 30, 31, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
pack_rows(NULL, 32, 33, bold = FALSE, latex_gap_space = '-2\\\\defaultaddspace',
          escape = FALSE, colnum = 0) |>
footnote(general = paste0('\\\\scriptsize The sample is users in the evaluation stage, $n = $ ',
                          prettyNum(eval_n, big.mark = ','),
                          ". Estimates are of subgroup means, produced from sample means and standard
),
escape = FALSE,
threeparttable = TRUE,
general_title = '')
```

Table 9: Composition of baseline vaccine attitude types.

	Missing	Hesitant	Undecided	Eager
Demographics				
Age	25.669	24.897	24.785	25.140
	(0.622)	(0.096)	(0.066)	(0.078)
% Non-male	0.669	0.485	0.482	0.458
	(0.040)	(0.007)	(0.005)	(0.006)
% Christian	0.838	0.792	0.784	0.765
	(0.032)	(0.006)	(0.004)	(0.005)
Religiosity (1:6)	3.382	3.798	3.952	3.983
	(0.193)	(0.024)	(0.016)	(0.019)
% Party aligned	0.110	0.265	0.269	0.374
	(0.027)	(0.006)	(0.004)	(0.006)
% Urban	0.493	0.595	0.594	0.538
	(0.043)	(0.007)	(0.005)	(0.006)
Socioeconomic status				
% Completed secondary school	0.937	0.909	0.923	0.897
	(0.022)	(0.004)	(0.003)	(0.004)
Household size	4.478	5.458	5.752	6.041
	(0.290)	(0.046)	(0.032)	(0.042)
Assets index (-5:5)	2.882	4.273	4.365	4.178
	(0.172)	(0.021)	(0.014)	(0.018)
% Employed	0.287	0.369	0.357	0.353
	(0.039)	(0.007)	(0.005)	(0.006)
Hours to nearest health facility	1.113	1.728	1.739	1.898
	(0.285)	(0.064)	(0.039)	(0.046)
Healthcare access/knowledge/attitudes				
COVID knowledge index (-5:5)	2.654	3.273	3.792	3.855
	(0.115)	(0.016)	(0.010)	(0.012)
% Know where to get COVID vaccine	0.673	0.679	0.719	0.760
	(0.046)	(0.007)	(0.005)	(0.005)
% Any prior vaccination	0.337	0.328	0.397	0.475
	(0.048)	(0.007)	(0.005)	(0.006)
Ease of getting COVID vaccine (-2:2)	3.027	2.964	2.957	3.042
	(0.114)	(0.016)	(0.010)	(0.014)
Trust index (-14:14)	0.581	-1.705	0.522	4.135
	(0.525)	(0.097)	(0.064)	(0.083)

The sample is users in the evaluation stage, $n = 22{,}052$. Estimates are of subgroup means, produced from sample means and standard error of the sample mean. Subgroups are defined on pre-treatment response to the vaccine willingness question; *hesitant* respondents reported that they wanted to get a vaccine "not at all;" *undecided* respondents wanted to get a vaccine "a little" or "moderately;" *eager* respondents wanted to get a vaccine "very much."

```
#####################################################
### code chunk number 26: hte_concern_calcs
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'

concerns <- c(`1` = 'Side effects',
```

```
              `2` = 'Vaccine does not work',
              `3` = 'COVID is not real',
              `4` = 'Protected by God',
              `5` = 'Do not trust healthcare workers',
              `6` = 'Do not trust government',
              `7` = 'Not sure what to believe')

# create data frames for figures
response_kenya <- df_kenya |>
  mutate(concern = concerns[concern_id1]) |>
  drop_na(concern, willingness, willingness_2, get_vaccinated, get_vaccinated_2) |>
  group_by(concern) |>
  summarise(willing_mean = mean(willingness_2, na.rm = TRUE),
            willing_sd = sd(willingness_2, na.rm = TRUE) / sqrt(sum(!is.na(willingness_2))),
            diff_willingness_mean = mean(willingness_2-willingness, na.rm = TRUE),
            diff_willingness_sd = sd(willingness_2-willingness, na.rm = TRUE) / sqrt(sum(!is.na(willingr
            intend_mean = mean(get_vaccinated_2, na.rm = TRUE),
            intend_sd = sd(get_vaccinated_2, na.rm = TRUE) / sqrt(sum(!is.na(get_vaccinated_2))),
            diff_intend_mean = mean(get_vaccinated_2-get_vaccinated, na.rm = TRUE),
            diff_intend_sd = sd(get_vaccinated_2-get_vaccinated, na.rm = TRUE) / sqrt(sum(!is.na(get_va
            n = n()) |>
  mutate(country = 'Kenya',
         pre_post = 'post',
         statistic_willing = diff_willingness_mean/diff_willingness_sd,
         statistic_intend = diff_intend_mean/diff_intend_sd) |>
  select(willing_mean, willing_sd, intend_mean, intend_sd, pre_post, concern, n, country,
         statistic_willing, statistic_intend)

response_pre_kenya <- df_kenya |>
  mutate(concern = concerns[concern_id1]) |>
  drop_na(concern, willingness, willingness_2, get_vaccinated, get_vaccinated_2) |>
  group_by(concern) |>
  summarise(willing_mean = mean(willingness, na.rm = TRUE),
            willing_sd = sd(willingness, na.rm = TRUE) /  sqrt(sum(!is.na(willingness))),
            intend_mean = mean(get_vaccinated, na.rm = TRUE),
            intend_sd = sd(get_vaccinated, na.rm = TRUE) /  sqrt(sum(!is.na(get_vaccinated))),
            n = n()) |>
  mutate(country = 'Kenya', pre_post = 'pre',
         statistic_willing = NA,
         statistic_intend = NA) |>
  select(willing_mean, willing_sd, intend_mean, intend_sd, pre_post, concern, n, country,
         statistic_willing, statistic_intend)


# nigeria dataset
response_nigeria <- df_nigeria |>
  mutate(concern = concerns[concern_id1]) |>
  drop_na(concern, willingness, willingness_2, get_vaccinated, get_vaccinated_2) |>
  group_by(concern) |>
  summarise(willing_mean = mean(willingness_2, na.rm = TRUE),
            willing_sd = sd(willingness_2, na.rm = TRUE) / sqrt(sum(!is.na(willingness_2))),
            diff_willingness_mean = mean(willingness_2-willingness, na.rm = TRUE),
            diff_willingness_sd = sd(willingness_2-willingness, na.rm = TRUE) / sqrt(sum(!is.na(willingr
```

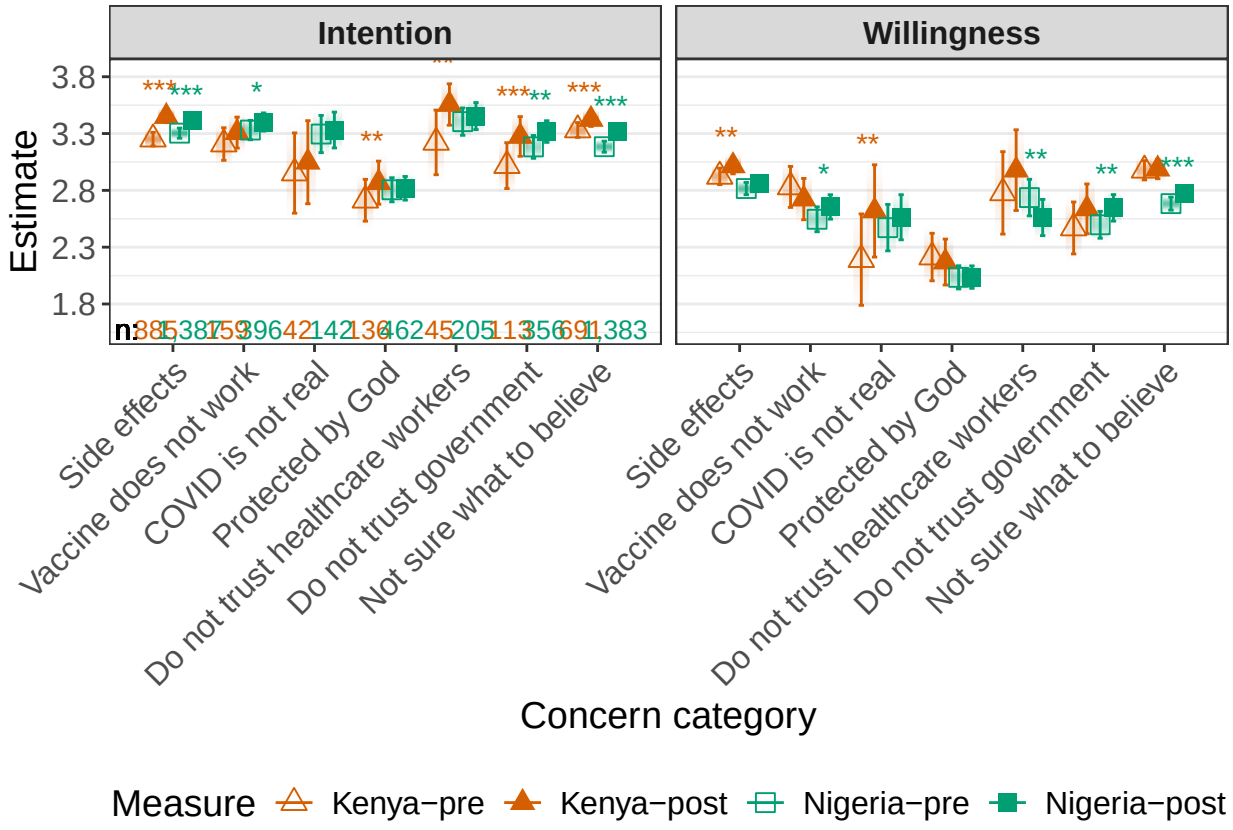
```
            intend_mean = mean(get_vaccinated_2, na.rm = TRUE),
            intend_sd = sd(get_vaccinated_2, na.rm = TRUE) / sqrt(sum(!is.na(get_vaccinated_2))),
            diff_intend_mean = mean(get_vaccinated_2-get_vaccinated, na.rm = TRUE),
            diff_intend_sd = sd(get_vaccinated_2-get_vaccinated, na.rm = TRUE) / sqrt(sum(!is.na(get_va
            n = n()) |>
  mutate(country = 'Nigeria',
         pre_post = 'post',
         statistic_willing = diff_willingness_mean/diff_willingness_sd,
         statistic_intend = diff_intend_mean/diff_intend_sd) |>
  select(willing_mean, willing_sd, intend_mean, intend_sd, pre_post, concern, n, country,
         statistic_willing, statistic_intend)

response_pre_nigeria <- df_nigeria |>
  mutate(concern = concerns[concern_id1]) |>
  drop_na(concern, willingness, willingness_2, get_vaccinated, get_vaccinated_2) |>
  group_by(concern) |>
  summarise(willing_mean = mean(willingness),
            willing_sd = sd(willingness, na.rm = TRUE) /  sqrt(sum(!is.na(willingness))),
            intend_mean = mean(get_vaccinated, na.rm = TRUE),
            intend_sd = sd(get_vaccinated, na.rm = TRUE) /  sqrt(sum(!is.na(get_vaccinated))),
            n = n()) |>
  mutate(country = 'Nigeria', pre_post = 'pre',
         statistic_willing = NA,
         statistic_intend = NA) |>
  select(willing_mean, willing_sd, intend_mean, intend_sd, pre_post, concern, n, country,
         statistic_willing, statistic_intend)


# combine datasets #
response_pre_post_both <- rbind(response_pre_kenya, response_kenya,
                                response_pre_nigeria, response_nigeria)


## reformat so intention and willingness are variable indicators for mean and sd cols
reshaped_data <- response_pre_post_both |>
  pivot_longer(cols = c(willing_mean, intend_mean, willing_sd, intend_sd),
               names_to = c('measure', '.value'),
               names_pattern = '(willing|intend)_(mean|sd)') |>
  mutate(pre_post = factor(pre_post, levels = c('pre','post')),
         concern = factor(concern, levels = concerns),
         country_time = factor(interaction(country, pre_post),
                               levels = c('Kenya.pre',
                                          'Kenya.post',
                                          'Nigeria.pre',
                                          'Nigeria.post')))

# add a column of stars indicating diffs that are stat. sig
reshaped_data <- reshaped_data |>
  mutate(significance = case_when(
    measure == "willing" & abs(statistic_willing) > 3.29 ~ '***', # 0.1% level
    measure == "willing" & abs(statistic_willing) > 2.58 ~ '**',  # 1% level
    measure == "willing" & abs(statistic_willing) > 1.96 ~ '*',   # 5% level
```

```
    measure == "intend" & abs(statistic_intend) > 3.29 ~ '***', # 0.1% level
    measure == "intend" & abs(statistic_intend) > 2.58 ~ '**',  # 1% level
    measure == "intend" & abs(statistic_intend) > 1.96 ~ '*',   # 5% level

    TRUE ~ NA_character_
  ))

# great var for height of highest confidence interval (to put * above)
reshaped_data <- reshaped_data |>
  group_by(country, concern, measure) |>
  mutate(ymax_star = max(mean + 1.96*sd)) |>
  ungroup()

# Filter for post data
post_data <- subset(reshaped_data,
                    pre_post == 'post' & measure == 'intend')

# Calculate a fixed y-position for the sample size labels based on the global min and max of the mean v
label_position <- min(reshaped_data$mean) - 0.5 * (max(reshaped_data$mean) - min(reshaped_data$mean))


# Plot
g <- ggplot(reshaped_data, aes(x = concern, y = mean,
                               color = country_time,
                               shape = country_time)) +
  stat_gradientinterval(aes(x = concern,
                            ydist = distributional::dist_normal(mean, sd),
                            fill = country_time),
                        width = 1,
                        position = position_dodge(0.75),
                        linewidth = 0,
                        point_size = 0,
                        point_alpha = 0,
                        interval_alpha = 0,
                        show.legend = FALSE,
                        fill_type = 'segments') +
  geom_point(position = position_dodge(0.75), size = 3) +  # Dodge position so points don't overlap
  geom_errorbar(aes(ymin = mean - 1.96*sd, ymax = mean + 1.96*sd),
                width = 0.3, position = position_dodge(0.75)) +
  geom_text(data = post_data, aes(label = prettyNum(n, big.mark = ','),
                                  y = label_position + .3,
                                  group = country,
                                  color =  country_time),
            position = position_dodge(.95), size = 3.5,
            show.legend = FALSE) +  # Display sample size for 'post'
  geom_text(data = post_data,
            aes(label = 'n:', x = .35, y = label_position + .3),
            color = 'black',
            size = 4,
            show.legend = FALSE) +
  facet_wrap(~ measure, ncol = 2, scales = 'fixed',
             labeller = labeller(measure = c(intend = 'Intention', willing = 'Willingness'))) +  # Fixe
```

```
  # add stars for significant diffs
  geom_text(aes(label = significance, y = ymax_star,
                group = country,
                color = country_time),
            position = position_dodge(0.75),
            vjust = -0.5,
            size = 4,
            show.legend = FALSE) +

  scale_shape_manual(name = 'Measure',
                     breaks = c('Kenya.pre','Kenya.post','Nigeria.pre','Nigeria.post'),
                     labels = c('Kenya-pre','Kenya-post','Nigeria-pre','Nigeria-post'),
                     values = c(2,17, 0, 15)) +  # Different shapes for country and pre/post
  scale_color_manual(name = 'Measure',
                     breaks = c('Kenya.pre','Kenya.post','Nigeria.pre','Nigeria.post'),
                     labels = c('Kenya-pre','Kenya-post','Nigeria-pre','Nigeria-post'),
                     values = c(cbPalette[7],cbPalette[7], cbPalette[4], cbPalette[4])) +
  scale_fill_manual(name = 'Measure',
                    breaks = c('Kenya.pre','Kenya.post','Nigeria.pre','Nigeria.post'),
                    labels = c('Kenya-pre','Kenya-post','Nigeria-pre','Nigeria-post'),
                    values = c(cbPalette[7],cbPalette[7], cbPalette[4], cbPalette[4]), guide = 'none') +

  scale_x_discrete(expand = c(0.15, 0)) +
  scale_y_continuous(breaks = seq(1.8, 4, by = .5)) +

  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 12),
    axis.line.x = NULL,
    axis.text.y = element_text(size = 12),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 12),
    strip.text = element_text(size = 20, face = 'bold'),
    legend.position = 'bottom'
  ) +
  labs(y = 'Estimate', x = 'Concern category', fill = 'Country', shape = 'Meaure') +
  theme(panel.grid.major.x = element_blank()) +
  vcf_theme()



#####################################################
### code chunk number 27: hte_concern
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| fig.align='center',
#| fig.width=11,
#| fig.height = 7,
#| prefix.string = 'fig',
#| fig.pos = '!ht',
#| fig.cap=
#| paste0('\\textbf{Among respondents in concern-addressing group, pre- and
```

```
#|  post-treatment COVID-19 vaccine intention and willingness.} ',
#| 'The sample is users in the concern-addressing condition in the evaluation
#|  stage who have complete pre- and post-treatment response measures and who
#|  provided information on their concerns, $n = $ ',
#|   prettyNum(sum(df_eval$treatment_group=='Concerns' &
#|                 !is.na(df_eval$willingness_2) &
#|                 !is.na(df_eval$concern_id1)), big.mark = ','),
#| '. Samples within each concern category by country are presented at the
#|  bottom of the Intention plot, color coded by country. ',
#| "Estimates are of subgroup means, produced from sample means and standard error of the sample mean.
#|  Subgroups are defined on users' first expressed concern.
#|  Error bars represent symmetric 95\\% confidence intervals.
#|  Stars represent statistical significance of the p-value associated with the
#|  difference in pre- and post-treatment mean response by primary concern
#|  category and country (*p < 0.05, **p < 0.01, ***p < 0.001)."),
#| strip.white=TRUE,
#| results='asis'
g
```

```
## Warning: Removed 41 rows containing missing values or values outside the scale range (`geom_text()`)
```




```
##################################################
### code chunk number 28: heterogeneity_estimation
##################################################
#| eval = TRUE,
#| cache = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
```

```
#| results='asis'


hetero_covariates_list<-c("is_male", "age_median","cov_know_median",
                          "health_access_median","party_aligned",
                          "religiosity_median","digital_index_median")


## response function (man/above median)
for (covariate in hetero_covariates_list){
  ## A) Difference-in-means estimate (i.e, treating missing data as random)
  est_DM <- paste0("est_DM_", covariate)
  est_formula <- paste0(est_DM, " <- estimatr::lm_robust(response ~ treatment_group, data = df_eval[whi
  eval(parse(text = est_formula))

  est_DM_alt <- paste0("est_DM_", covariate, "_alt")
  est_formula_alt <- paste0(est_DM_alt, " <- estimatr::lm_robust(response ~ treatment_group, data = df_
  eval(parse(text = est_formula_alt))

  ## B) Imputing pre-test as post-test
  est_IMP <- paste0("est_IMP_", covariate)
  est_formula_IMP <- paste0(est_IMP, " <- estimatr::lm_robust(response_imputed ~ treatment_group, data =
  eval(parse(text = est_formula_IMP))

  est_IMP_alt <- paste0("est_IMP_", covariate, "_alt")
  est_formula_IMP_alt <- paste0(est_IMP_alt, " <- estimatr::lm_robust(response_imputed ~ treatment_grou
  eval(parse(text = est_formula_IMP_alt))

  ## C) Generalized random forest
  est_GRF <- paste0("est_GRF_", covariate)
  est_formula_GRF <- paste0(est_GRF, " <- mcf_estimate(df_eval[which(df_eval$", covariate, " == 1),], '
  eval(parse(text = est_formula_GRF))

  est_GRF_alt <- paste0("est_GRF_", covariate, "_alt")
  est_formula_GRF_alt <- paste0(est_GRF_alt, " <- mcf_estimate(df_eval_alt[which(df_eval_alt$", covariat
  eval(parse(text = est_formula_GRF_alt))

  ## D) IP Weighting
  est_IP <- paste0("est_IP_", covariate)
  est_formula_IP <- paste0(est_IP, " <- IP_estimate(df_eval[which(df_eval$", covariate, " == 1),], cova
  eval(parse(text = est_formula_IP))

  est_IP_alt <- paste0("est_IP_", covariate, "_alt")
  est_formula_IP_alt <- paste0(est_IP_alt, " <- IP_estimate(df_eval_alt[which(df_eval_alt$", covariate,
  eval(parse(text = est_formula_IP_alt))

  ## E) Lin estimator
  est_Lin <- paste0("est_Lin_", covariate)
  if (covariate %in% c("health_access_median","party_aligned","age_median")){
    covariate_lin_temp <- as.formula(paste('~', paste0(
      c('willingness_imputed', 'get_vaccinated_imputed'),
      collapse = ' + ')))
  }else {
```

```
    covariate_lin_temp<-covariate_lin
  }

  est_formula_Lin <- paste0(est_Lin, " <- estimatr::lm_lin(response ~ treatment_group, data = df_eval[wh
  eval(parse(text = est_formula_Lin))

  est_Lin_alt <- paste0("est_Lin_", covariate, "_alt")
  est_formula_Lin_alt <- paste0(est_Lin_alt, " <- estimatr::lm_lin(response ~ treatment_group, data = df
  eval(parse(text = est_formula_Lin_alt))
}


## response function (no man/below median)
for (covariate in hetero_covariates_list){

  ## A) Difference-in-means estimate (i.e, treating missing data as random)
  est_DM <- paste0("est_DM_no", covariate)
  est_formula <- paste0(est_DM, " <- estimatr::lm_robust(response ~ treatment_group, data = df_eval[whi
  eval(parse(text = est_formula))

  est_DM_alt <- paste0("est_DM_no", covariate, "_alt")
  est_formula_alt <- paste0(est_DM_alt, " <- estimatr::lm_robust(response ~ treatment_group, data = df_
  eval(parse(text = est_formula_alt))

  ## B) Imputing pre-test as post-test
  est_IMP <- paste0("est_IMP_no", covariate)
  est_formula_IMP <- paste0(est_IMP, " <- estimatr::lm_robust(response_imputed ~ treatment_group, data =
  eval(parse(text = est_formula_IMP))

  est_IMP_alt <- paste0("est_IMP_no", covariate, "_alt")
  est_formula_IMP_alt <- paste0(est_IMP_alt, " <- estimatr::lm_robust(response_imputed ~ treatment_group
  eval(parse(text = est_formula_IMP_alt))

  ## C) Generalized random forest
  est_GRF <- paste0("est_GRF_no", covariate)
  est_formula_GRF <- paste0(est_GRF, " <- mcf_estimate(df_eval[which(df_eval$", covariate, " != 1),], '
  eval(parse(text = est_formula_GRF))

  est_GRF_alt <- paste0("est_GRF_no", covariate, "_alt")
  est_formula_GRF_alt <- paste0(est_GRF_alt, " <- mcf_estimate(df_eval_alt[which(df_eval_alt$", covariat
  eval(parse(text = est_formula_GRF_alt))

  ## D) IP Weighting
  est_IP <- paste0("est_IP_no", covariate)
  est_formula_IP <- paste0(est_IP, " <- IP_estimate(df_eval[which(df_eval$", covariate, " != 1),], covar
  eval(parse(text = est_formula_IP))

  est_IP_alt <- paste0("est_IP_no", covariate, "_alt")
  est_formula_IP_alt <- paste0(est_IP_alt, " <- IP_estimate(df_eval_alt[which(df_eval_alt$", covariate,
  eval(parse(text = est_formula_IP_alt))


  ## E) Lin estimator
```

```
  est_Lin <- paste0("est_Lin_no", covariate)
  est_formula_Lin <- paste0(est_Lin, " <- estimatr::lm_lin(response ~ treatment_group, data = df_eval[wh
  eval(parse(text = est_formula_Lin))

  est_Lin_alt <- paste0("est_Lin_no", covariate, "_alt")
  est_formula_Lin_alt <- paste0(est_Lin_alt, " <- estimatr::lm_lin(response ~ treatment_group, data = di
  eval(parse(text = est_formula_Lin_alt))
}



#####################################################
### code chunk number 29: heterogeneity_results
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'


hetero_covariates_list<-c("is_male", "age_median","cov_know_median",
                          "health_access_median","party_aligned",
                          "religiosity_median","digital_index_median")

covariate_terms <- list(
  is_male = c("Male", "Non-male"),
  age_median = c("Age above median", "Age below median"),
  cov_know_median = c("Scientific knowledge above median", "Scientific knowledge below median"),
  health_access_median = c("Health access above median", "Health access below median"),
  party_aligned = c("Aligned with governing party", "Not aligned with governing party"),
  religiosity_median = c("Religiosity above median", "Religiosity below median"),
  digital_index_median = c("Digital index score above median", "Digital index score below median")
)

table_terms <- list(
  is_male = c("gender"),
  age_median = c("age"),
  cov_know_median = c("scientific knowledge"),
  health_access_median = c("health access"),
  party_aligned = c("alignment with governing party"),
  religiosity_median = c("religiosity"),
  digital_index_median = c("digital index score")
)


for (covariate in hetero_covariates_list) {

  temp_list <- list(
    DM = get(paste0("est_DM_", covariate)),
    IMP = get(paste0("est_IMP_", covariate)),
    Lin = get(paste0("est_Lin_", covariate)),
    IP = get(paste0("est_IP_", covariate))
  )
```

```r
assign(paste0("out_list_", covariate), temp_list)

temp_list <- list(
  DM = get(paste0("est_DM_no", covariate)),
  IMP = get(paste0("est_IMP_no", covariate)),
  Lin = get(paste0("est_Lin_no", covariate)),
  IP = get(paste0("est_IP_no", covariate))
)

assign(paste0("out_list_no", covariate), temp_list)


temp_list <- list(
  DM = compare_models(get(paste0("est_DM_no", covariate)), get(paste0("est_DM_", covariate)))$Differe
  IMP = compare_models(get(paste0("est_IMP_no", covariate)), get(paste0("est_IMP_", covariate)))$Diff
  Lin = compare_models(get(paste0("est_Lin_no", covariate)), get(paste0("est_Lin_", covariate)))$Diff
  IP = compare_models(get(paste0("est_IP_no", covariate)), get(paste0("est_IP_", covariate)))$Differe
)
assign(paste0("out_list_diff_", covariate), temp_list)

temp <- list(
  DM = compare_models(get(paste0("est_DM_no", covariate, "_alt")), get(paste0("est_DM_", covariate, "
  IMP = compare_models(get(paste0("est_IMP_no", covariate, "_alt")), get(paste0("est_IMP_", covariate
  Lin = compare_models(get(paste0("est_Lin_no", covariate, "_alt")), get(paste0("est_Lin_", covariate
  IP = compare_models(get(paste0("est_IP_no", covariate, "_alt")), get(paste0("est_IP_", covariate, "
)
assign(paste0("out_list_diff_alt_", covariate), temp_list)

out_list_covariate <- modelsummary(get(paste0('out_list_', covariate)),
                                   output = 'modelsummary_list')
out_list_no_covariate <- modelsummary(get(paste0('out_list_no', covariate)),
                                      output = 'modelsummary_list')

out_list_diff<-get(paste0('out_list_diff_', covariate))
out_list_diff_alt<-get(paste0('out_list_diff_alt_', covariate))

for(x in names(out_list_covariate)){
  # add on Concerns - PSA coefficients

  out_list_covariate[[x]]$tidy <- rbind(out_list_covariate[[x]]$tidy |>
                                          select(term, estimate, std.error,
                                                 statistic, p.value),
                                        broom::tidy(get(paste0('est_', x, '_', covariate, '_alt')))[3
                                          mutate(term = 'treatment_groupConcerns2') |>
                                          select(term, estimate, std.error, statistic,p.value))

  out_list_no_covariate[[x]]$tidy <- rbind(out_list_no_covariate[[x]]$tidy |>
                                             select(term, estimate, std.error,
                                                    statistic, p.value),
                                           broom::tidy(get(paste0('est_', x, '_no', covariate, '_alt'
                                             mutate(term = 'treatment_groupConcerns2') |>
                                             select(term, estimate, std.error, statistic, p.value))
```

```
  out_list_diff[[x]]$tidy <- rbind(out_list_diff[[x]]$tidy |>
                                     select(term, estimate, std.error,
                                            statistic,p.value),
                                   out_list_diff_alt[[x]]$tidy[which(out_list_diff_alt[[x]]$tidy$term
                                     mutate(term = 'treatment_groupConcerns2') |>
                                     select(term, estimate, std.error,
                                            statistic, p.value))

  # drop p-values for control condition so no stars
  out_list_covariate[[x]]$tidy[which(out_list_covariate[[x]]$tidy$term == '(Intercept)'), c('p.value'
  out_list_no_covariate[[x]]$tidy[which(out_list_no_covariate[[x]]$tidy$term == '(Intercept)'), c('p.
}

# Add GRF; different format

out_list_covariate[['GRF']] <- format_grf_summary(get(paste0('est_GRF_', covariate)))
out_list_covariate[['GRF']]$tidy <- rbind(out_list_covariate[['GRF']]$tidy,format_grf_summary(get(pas
                                          mutate(term = 'treatment_groupConcerns2'))
out_list_covariate[['GRF']]$tidy[which(out_list_covariate[['GRF']]$tidy$term == '(Intercept)'),
                                 c('p.value', 'statistic')] <- NA

# Models without the covariate
out_list_no_covariate[['GRF']] <- format_grf_summary(get(paste0('est_GRF_no', covariate)))
out_list_no_covariate[['GRF']]$tidy <- rbind(out_list_no_covariate[['GRF']]$tidy,format_grf_summary(g
                                             mutate(term = 'treatment_groupConcerns2'))
out_list_no_covariate[['GRF']]$tidy[which(out_list_no_covariate[['GRF']]$tidy$term == '(Intercept)'),
                                    c('p.value', 'statistic')] <- NA

# Difference between with and without the covariate
out_list_diff[['GRF']] <- compare_models(format_grf_summary(get(paste0('est_GRF_no', covariate))), fo

out_list_diff[['GRF']]$tidy <- rbind(out_list_diff[['GRF']]$tidy,
                                     compare_models(format_grf_summary(get(paste0('est_GRF_no', covar
                                                    format_grf_summary(get(paste0('est_GRF_', covaria
                                       filter(term == 'treatment_groupConcerns') |>
                                       mutate(term = 'treatment_groupConcerns2'))

named_list <- list()
named_list[[covariate_terms[[covariate]][1]]] <- out_list_covariate
named_list[[covariate_terms[[covariate]][2]]] <- out_list_no_covariate
named_list[["Difference"]] <- out_list_diff


print(modelsummary(named_list,
                   shape= 'rbind',
                   output = 'latex',
                   coef_map = c('treatment_groupPSA' = 'PSA - Control',
                                'treatment_groupConcerns' = 'Concerns - Control',
                                'treatment_groupConcerns2' = 'Concerns - PSA',
                                '(Intercept)' = 'Control mean'),
                   stars = TRUE ,
                   gof_map = list(list('raw' = 'nobs',
                                       'clean' = 'n',
```

```
                                          'fmt' = f1)),
                    escape = FALSE,
                    caption= paste0("Treatment effect estimates and response under alternative approach
          kable_styling(latex_options = c('HOLD_position')) |>
          footnote(general = paste0('\\\\footnotesize The sample is users in the evaluation stage. The
          ),
          escape = FALSE,
          threeparttable = TRUE,
          general_title = '')
  )
}
```

```
## 3 coefficients  not defined because the design matrix is rank deficient
## 3 coefficients  not defined because the design matrix is rank deficient
##
## 3 coefficients  not defined because the design matrix is rank deficient
##
## 3 coefficients  not defined because the design matrix is rank deficient
##
## 3 coefficients  not defined because the design matrix is rank deficient
##
## 3 coefficients  not defined because the design matrix is rank deficient
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
## \begin{table}[H]
## \centering\centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lccccc}
## \toprule
##   & DM & IMP & Lin & IP & GRF\\
## \midrule
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Male}}\\
## \midrule \hspace{1em}PSA - Control & 0.055 & 0.040 & 0.029 & 0.043 & 0.041+\\
## \hspace{1em} & (0.041) & (0.041) & (0.021) & (0.041) & (0.021)\\
## \hspace{1em}Concerns - Control & 0.295*** & 0.266*** & 0.189*** & 0.256*** & 0.204***\\
## \hspace{1em} & (0.040) & (0.039) & (0.023) & (0.040) & (0.024)\\
## \hspace{1em}Concerns - PSA & 0.240*** & 0.226*** & 0.161*** & 0.213*** & 0.166***\\
## \hspace{1em} & (0.040) & (0.039) & (0.025) & (0.040) & (0.025)\\
## \hspace{1em}Control mean & -0.048 & -0.042 & -0.002 & -0.030 & -0.008\\
## \hspace{1em} & (0.029) & (0.029) & (0.014) & (0.029) & (0.015)\\
## \hspace{1em}n & 11,072 & 11,542 & 11,072 & 11,072 & 11,072\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Non-male}}\\
## \midrule \hspace{1em}PSA - Control & 0.080+ & 0.072+ & 0.039+ & 0.073 & 0.045*\\
## \hspace{1em} & (0.045) & (0.043) & (0.022) & (0.045) & (0.022)\\
```

```
## \hspace{1em}Concerns - Control & 0.354*** & 0.314*** & 0.255*** & 0.329*** & 0.268***\\
## \hspace{1em} & (0.045) & (0.043) & (0.025) & (0.045) & (0.026)\\
## \hspace{1em}Concerns - PSA & 0.274*** & 0.243*** & 0.216*** & 0.256*** & 0.224***\\
## \hspace{1em} & (0.044) & (0.042) & (0.026) & (0.044) & (0.027)\\
## \hspace{1em}Control mean & -0.207 & -0.196 & -0.159 & -0.203 & -0.166\\
## \hspace{1em} & (0.032) & (0.031) & (0.014) & (0.033) & (0.016)\\
## \hspace{1em}n & 9,753 & 10,510 & 9,753 & 9,753 & 9,753\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Difference}}\\
## \midrule \hspace{1em}PSA - Control & -0.024 & -0.032 & -0.010 & -0.030 & -0.005\\
## \hspace{1em} & (0.061) & (0.059) & (0.030) & (0.061) & (0.030)\\
## \hspace{1em}Concerns - Control & -0.059 & -0.048 & -0.065+ & -0.073 & -0.063+\\
## \hspace{1em} & (0.060) & (0.058) & (0.034) & (0.061) & (0.035)\\
## \hspace{1em}Concerns - PSA & -0.059 & -0.048 & -0.065+ & -0.073 & -0.057\\
## \hspace{1em} & (0.060) & (0.058) & (0.034) & (0.061) & (0.037)\\
## \hspace{1em}Control mean & 0.160*** & 0.154*** & 0.157*** & 0.173*** & 0.157***\\
## \hspace{1em} & (0.043) & (0.042) & (0.020) & (0.044) & (0.022)\\
## \bottomrule
## \multicolumn{6}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is the combine
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
```

```
## \begin{table}[H]
## \centering\centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lccccc}
## \toprule
##   & DM & IMP & Lin & IP & GRF\\
## \midrule
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Age above median}}\\
## \midrule \hspace{1em}PSA - Control & -0.026 & -0.034 & 0.037 & 0.009 & 0.041\\
## \hspace{1em} & (0.081) & (0.079) & (0.036) & (0.081) & (0.037)\\
## \hspace{1em}Concerns - Control & 0.255** & 0.207** & 0.235*** & 0.256** & 0.244***\\
## \hspace{1em} & (0.079) & (0.076) & (0.043) & (0.080) & (0.045)\\
## \hspace{1em}Concerns - PSA & 0.281*** & 0.241** & 0.198*** & 0.250** & 0.203***\\
## \hspace{1em} & (0.078) & (0.075) & (0.045) & (0.078) & (0.045)\\
## \hspace{1em}Control mean & -0.136 & -0.133 & -0.149 & -0.152 & -0.154\\
## \hspace{1em} & (0.058) & (0.056) & (0.024) & (0.058) & (0.030)\\
## \hspace{1em}n & 3,037 & 3,273 & 3,037 & 3,037 & 3,037\\
## \addlinespace[0.5em]
```

```
## \multicolumn{6}{l}{\textit{Age below median}}\\
## \midrule \hspace{1em}PSA - Control & 0.081* & 0.068* & 0.034* & 0.053 & 0.039*\\
## \hspace{1em} & (0.033) & (0.032) & (0.016) & (0.033) & (0.017)\\
## \hspace{1em}Concerns - Control & 0.336*** & 0.304*** & 0.222*** & 0.283*** & 0.231***\\
## \hspace{1em} & (0.032) & (0.031) & (0.019) & (0.032) & (0.019)\\
## \hspace{1em}Concerns - PSA & 0.255*** & 0.236*** & 0.187*** & 0.229*** & 0.192***\\
## \hspace{1em} & (0.032) & (0.031) & (0.020) & (0.032) & (0.020)\\
## \hspace{1em}Control mean & -0.120 & -0.111 & -0.063 & -0.094 & -0.070\\
## \hspace{1em} & (0.023) & (0.023) & (0.011) & (0.023) & (0.012)\\
## \hspace{1em}n & 17,788 & 18,779 & 17,788 & 17,788 & 17,788\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Difference}}\\
## \midrule \hspace{1em}PSA - Control & -0.107 & -0.102 & 0.002 & -0.044 & 0.002\\
## \hspace{1em} & (0.087) & (0.085) & (0.039) & (0.087) & (0.040)\\
## \hspace{1em}Concerns - Control & -0.081 & -0.097 & 0.013 & -0.027 & 0.013\\
## \hspace{1em} & (0.086) & (0.082) & (0.047) & (0.087) & (0.049)\\
## \hspace{1em}Concerns - PSA & -0.081 & -0.097 & 0.013 & -0.027 & 0.011\\
## \hspace{1em} & (0.086) & (0.082) & (0.047) & (0.087) & (0.050)\\
## \hspace{1em}Control mean & -0.016 & -0.022 & -0.086** & -0.058 & -0.084**\\
## \hspace{1em} & (0.062) & (0.061) & (0.026) & (0.063) & (0.032)\\
## \bottomrule
## \multicolumn{6}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is the combine
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
```

```
## \begin{table}[H]
## \centering\centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lccccc}
## \toprule
##   & DM & IMP & Lin & IP & GRF\\
## \midrule
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Scientific knowledge above median}}\\
## \midrule \hspace{1em}PSA - Control & 0.096* & 0.079* & 0.029+ & 0.064+ & 0.037*\\
## \hspace{1em} & (0.037) & (0.037) & (0.018) & (0.037) & (0.018)\\
## \hspace{1em}Concerns - Control & 0.341*** & 0.323*** & 0.229*** & 0.293*** & 0.240***\\
## \hspace{1em} & (0.036) & (0.036) & (0.021) & (0.037) & (0.021)\\
## \hspace{1em}Concerns - PSA & 0.246*** & 0.244*** & 0.199*** & 0.230*** & 0.198***\\
## \hspace{1em} & (0.036) & (0.035) & (0.021) & (0.036) & (0.022)\\
## \hspace{1em}Control mean & 0.090 & 0.099 & 0.154 & 0.118 & 0.147\\
```

```
## \hspace{1em} & (0.027) & (0.027) & (0.012) & (0.027) & (0.014)\\
## \hspace{1em}n & 12,196 & 12,750 & 12,196 & 12,196 & 12,196\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Scientific knowledge below median}}\\
## \midrule \hspace{1em}PSA - Control & 0.010 & 0.006 & 0.037 & 0.035 & 0.053*\\
## \hspace{1em} & (0.049) & (0.048) & (0.026) & (0.049) & (0.026)\\
## \hspace{1em}Concerns - Control & 0.279*** & 0.233*** & 0.210*** & 0.266*** & 0.231***\\
## \hspace{1em} & (0.049) & (0.047) & (0.030) & (0.049) & (0.030)\\
## \hspace{1em}Concerns - PSA & 0.269*** & 0.227*** & 0.173*** & 0.232*** & 0.179***\\
## \hspace{1em} & (0.049) & (0.047) & (0.031) & (0.049) & (0.032)\\
## \hspace{1em}Control mean & -0.414 & -0.401 & -0.397 & -0.420 & -0.409\\
## \hspace{1em} & (0.035) & (0.034) & (0.017) & (0.035) & (0.018)\\
## \hspace{1em}n & 8,629 & 9,302 & 8,629 & 8,629 & 8,629\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Difference}}\\
## \midrule \hspace{1em}PSA - Control & 0.086 & 0.073 & -0.008 & 0.029 & -0.016\\
## \hspace{1em} & (0.062) & (0.060) & (0.031) & (0.062) & (0.032)\\
## \hspace{1em}Concerns - Control & 0.062 & 0.089 & 0.019 & 0.027 & 0.009\\
## \hspace{1em} & (0.061) & (0.059) & (0.036) & (0.061) & (0.036)\\
## \hspace{1em}Concerns - PSA & 0.062 & 0.089 & 0.019 & 0.027 & 0.019\\
## \hspace{1em} & (0.061) & (0.059) & (0.036) & (0.061) & (0.038)\\
## \hspace{1em}Control mean & 0.504*** & 0.501*** & 0.551*** & 0.538*** & 0.557***\\
## \hspace{1em} & (0.044) & (0.043) & (0.021) & (0.044) & (0.022)\\
## \bottomrule
## \multicolumn{6}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is the combine
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
```

```
## \begin{table}[H]
## \centering\centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lccccc}
## \toprule
##   & DM & IMP & Lin & IP & GRF\\
## \midrule
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Health access above median}}\\
## \midrule \hspace{1em}PSA - Control & 0.111* & 0.106* & 0.051+ & 0.088+ & 0.055+\\
## \hspace{1em} & (0.053) & (0.053) & (0.028) & (0.053) & (0.028)\\
## \hspace{1em}Concerns - Control & 0.264*** & 0.263*** & 0.183*** & 0.222*** & 0.193***\\
## \hspace{1em} & (0.053) & (0.053) & (0.032) & (0.054) & (0.033)\\
```

```
## \hspace{1em}Concerns - PSA & 0.153** & 0.157** & 0.132*** & 0.135* & 0.132***\\
## \hspace{1em} & (0.053) & (0.052) & (0.033) & (0.053) & (0.034)\\
## \hspace{1em}Control mean & 0.146 & 0.153 & 0.195 & 0.166 & 0.192\\
## \hspace{1em} & (0.038) & (0.038) & (0.019) & (0.038) & (0.019)\\
## \hspace{1em}n & 5,801 & 5,874 & 5,801 & 5,801 & 5,801\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Health access below median}}\\
## \midrule \hspace{1em}PSA - Control & 0.043 & 0.031 & 0.023 & 0.034 & 0.031+\\
## \hspace{1em} & (0.036) & (0.035) & (0.018) & (0.036) & (0.018)\\
## \hspace{1em}Concerns - Control & 0.352*** & 0.313*** & 0.239*** & 0.299*** & 0.250***\\
## \hspace{1em} & (0.036) & (0.034) & (0.020) & (0.036) & (0.021)\\
## \hspace{1em}Concerns - PSA & 0.309*** & 0.281*** & 0.216*** & 0.267*** & 0.216***\\
## \hspace{1em} & (0.035) & (0.034) & (0.021) & (0.036) & (0.022)\\
## \hspace{1em}Control mean & -0.227 & -0.217 & -0.179 & -0.206 & -0.185\\
## \hspace{1em} & (0.026) & (0.025) & (0.011) & (0.026) & (0.013)\\
## \hspace{1em}n & 15,024 & 16,178 & 15,024 & 15,024 & 15,024\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Difference}}\\
## \midrule \hspace{1em}PSA - Control & 0.069 & 0.075 & 0.028 & 0.054 & 0.023\\
## \hspace{1em} & (0.065) & (0.064) & (0.033) & (0.064) & (0.033)\\
## \hspace{1em}Concerns - Control & -0.088 & -0.050 & -0.056 & -0.076 & -0.057\\
## \hspace{1em} & (0.064) & (0.063) & (0.038) & (0.065) & (0.039)\\
## \hspace{1em}Concerns - PSA & -0.088 & -0.050 & -0.056 & -0.076 & -0.084*\\
## \hspace{1em} & (0.064) & (0.063) & (0.038) & (0.065) & (0.040)\\
## \hspace{1em}Control mean & 0.373*** & 0.371*** & 0.374*** & 0.372*** & 0.376***\\
## \hspace{1em} & (0.046) & (0.046) & (0.022) & (0.046) & (0.023)\\
## \bottomrule
## \multicolumn{6}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is the combine
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
## 3 coefficients  not defined because the design matrix is rank deficient
##
## 3 coefficients  not defined because the design matrix is rank deficient
##
## 3 coefficients  not defined because the design matrix is rank deficient
```

```
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
```

```
## \begin{table}[H]
## \centering\centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lccccc}
```

```
## \toprule
##   & DM & IMP & Lin & IP & GRF\\
## \midrule
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Aligned with governing party}}\\
## \midrule \hspace{1em}PSA - Control & 0.009 & 0.008 & 0.018 & 0.005 & 0.034\\
## \hspace{1em} & (0.052) & (0.051) & (0.028) & (0.052) & (0.028)\\
## \hspace{1em}Concerns - Control & 0.247*** & 0.225*** & 0.155*** & 0.215*** & 0.165***\\
## \hspace{1em} & (0.049) & (0.048) & (0.031) & (0.050) & (0.031)\\
## \hspace{1em}Concerns - PSA & 0.238*** & 0.217*** & 0.137*** & 0.209*** & 0.131***\\
## \hspace{1em} & (0.049) & (0.048) & (0.031) & (0.050) & (0.032)\\
## \hspace{1em}Control mean & 0.247 & 0.251 & 0.278 & 0.259 & 0.268\\
## \hspace{1em} & (0.036) & (0.036) & (0.019) & (0.037) & (0.018)\\
## \hspace{1em}n & 6,340 & 6,618 & 6,340 & 6,340 & 6,340\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Not aligned with governing party}}\\
## \midrule \hspace{1em}PSA - Control & 0.085* & 0.070+ & 0.040* & 0.064+ & 0.043*\\
## \hspace{1em} & (0.037) & (0.036) & (0.018) & (0.037) & (0.018)\\
## \hspace{1em}Concerns - Control & 0.353*** & 0.313*** & 0.252*** & 0.311*** & 0.260***\\
## \hspace{1em} & (0.037) & (0.035) & (0.021) & (0.037) & (0.021)\\
## \hspace{1em}Concerns - PSA & 0.268*** & 0.243*** & 0.212*** & 0.248*** & 0.217***\\
## \hspace{1em} & (0.036) & (0.035) & (0.022) & (0.037) & (0.022)\\
## \hspace{1em}Control mean & -0.281 & -0.269 & -0.229 & -0.263 & -0.235\\
## \hspace{1em} & (0.026) & (0.026) & (0.011) & (0.026) & (0.014)\\
## \hspace{1em}n & 14,485 & 15,434 & 14,485 & 14,485 & 14,485\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Difference}}\\
## \midrule \hspace{1em}PSA - Control & -0.076 & -0.062 & -0.022 & -0.059 & -0.009\\
## \hspace{1em} & (0.063) & (0.062) & (0.033) & (0.063) & (0.033)\\
## \hspace{1em}Concerns - Control & -0.106+ & -0.088 & -0.096** & -0.096 & -0.095*\\
## \hspace{1em} & (0.061) & (0.060) & (0.037) & (0.062) & (0.037)\\
## \hspace{1em}Concerns - PSA & -0.106+ & -0.088 & -0.096** & -0.096 & -0.086*\\
## \hspace{1em} & (0.061) & (0.060) & (0.037) & (0.062) & (0.039)\\
## \hspace{1em}Control mean & 0.527*** & 0.520*** & 0.507*** & 0.521*** & 0.503***\\
## \hspace{1em} & (0.045) & (0.044) & (0.022) & (0.045) & (0.022)\\
## \bottomrule
## \multicolumn{6}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is the combine
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
## 3 coefficients  not defined because the design matrix is rank deficient
##
## 3 coefficients  not defined because the design matrix is rank deficient
##
## 3 coefficients  not defined because the design matrix is rank deficient
```

```
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
```

```
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
```

```
## \begin{table}[H]
## \centering\centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lccccc}
## \toprule
##   & DM & IMP & Lin & IP & GRF\\
## \midrule
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Religiosity above median}}\\
## \midrule \hspace{1em}PSA - Control & 0.106** & 0.094* & 0.034+ & 0.083* & 0.044*\\
## \hspace{1em} & (0.039) & (0.038) & (0.019) & (0.039) & (0.019)\\
## \hspace{1em}Concerns - Control & 0.347*** & 0.304*** & 0.231*** & 0.303*** & 0.244***\\
## \hspace{1em} & (0.039) & (0.038) & (0.022) & (0.039) & (0.022)\\
## \hspace{1em}Concerns - PSA & 0.241*** & 0.209*** & 0.197*** & 0.220*** & 0.200***\\
## \hspace{1em} & (0.038) & (0.037) & (0.023) & (0.039) & (0.023)\\
## \hspace{1em}Control mean & -0.137 & -0.128 & -0.072 & -0.116 & -0.081\\
## \hspace{1em} & (0.028) & (0.028) & (0.012) & (0.028) & (0.014)\\
## \hspace{1em}n & 12,804 & 13,596 & 12,804 & 12,804 & 12,804\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Religiosity below median}}\\
## \midrule \hspace{1em}PSA - Control & 0.000 & -0.013 & 0.037 & 0.001 & 0.030\\
## \hspace{1em} & (0.047) & (0.047) & (0.024) & (0.048) & (0.025)\\
## \hspace{1em}Concerns - Control & 0.287*** & 0.265*** & 0.209*** & 0.255*** & 0.215***\\
## \hspace{1em} & (0.046) & (0.045) & (0.028) & (0.047) & (0.028)\\
## \hspace{1em}Concerns - PSA & 0.288*** & 0.278*** & 0.173*** & 0.253*** & 0.183***\\
## \hspace{1em} & (0.047) & (0.045) & (0.029) & (0.047) & (0.029)\\
## \hspace{1em}Control mean & -0.099 & -0.092 & -0.081 & -0.090 & -0.081\\
## \hspace{1em} & (0.033) & (0.033) & (0.016) & (0.034) & (0.017)\\
## \hspace{1em}n & 8,021 & 8,456 & 8,021 & 8,021 & 8,021\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Difference}}\\
## \midrule \hspace{1em}PSA - Control & 0.106+ & 0.107+ & -0.002 & 0.082 & 0.013\\
## \hspace{1em} & (0.062) & (0.060) & (0.031) & (0.062) & (0.031)\\
## \hspace{1em}Concerns - Control & 0.059 & 0.038 & 0.022 & 0.048 & 0.029\\
## \hspace{1em} & (0.060) & (0.059) & (0.035) & (0.061) & (0.036)\\
## \hspace{1em}Concerns - PSA & 0.059 & 0.038 & 0.022 & 0.048 & 0.017\\
## \hspace{1em} & (0.060) & (0.059) & (0.035) & (0.061) & (0.038)\\
## \hspace{1em}Control mean & -0.038 & -0.036 & 0.010 & -0.025 & 0.001\\
## \hspace{1em} & (0.044) & (0.043) & (0.020) & (0.044) & (0.022)\\
## \bottomrule
## \multicolumn{6}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is the combine
## \end{tablenotes}
## \end{threeparttable}
## \end{table}
```

```
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
```

```
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
```

```
## \begin{table}[H]
## \centering\centering
## \caption{\label{tab:unnamed-chunk-1}Treatment effect estimates and response under alternative approa
## \centering
## \begin{threeparttable}
## \begin{tabular}[t]{lccccc}
## \toprule
##   & DM & IMP & Lin & IP & GRF\\
## \midrule
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Digital index score above median}}\\
## \midrule \hspace{1em}PSA - Control & 0.093* & 0.078* & 0.047* & 0.079* & 0.054**\\
## \hspace{1em} & (0.037) & (0.037) & (0.019) & (0.037) & (0.019)\\
## \hspace{1em}Concerns - Control & 0.350*** & 0.317*** & 0.249*** & 0.318*** & 0.261***\\
## \hspace{1em} & (0.037) & (0.036) & (0.022) & (0.037) & (0.022)\\
## \hspace{1em}Concerns - PSA & 0.257*** & 0.238*** & 0.203*** & 0.239*** & 0.205***\\
## \hspace{1em} & (0.037) & (0.036) & (0.023) & (0.037) & (0.023)\\
## \hspace{1em}Control mean & -0.144 & -0.134 & -0.093 & -0.131 & -0.099\\
## \hspace{1em} & (0.027) & (0.026) & (0.013) & (0.027) & (0.014)\\
## \hspace{1em}n & 12,721 & 13,440 & 12,721 & 12,721 & 12,721\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Digital index score below median}}\\
## \midrule \hspace{1em}PSA - Control & 0.020 & 0.013 & 0.010 & 0.020 & 0.017\\
## \hspace{1em} & (0.051) & (0.050) & (0.024) & (0.051) & (0.024)\\
## \hspace{1em}Concerns - Control & 0.282*** & 0.245*** & 0.176*** & 0.243*** & 0.191***\\
## \hspace{1em} & (0.049) & (0.048) & (0.027) & (0.050) & (0.028)\\
## \hspace{1em}Concerns - PSA & 0.262*** & 0.233*** & 0.166*** & 0.227*** & 0.175***\\
## \hspace{1em} & (0.050) & (0.048) & (0.029) & (0.050) & (0.030)\\
## \hspace{1em}Control mean & -0.088 & -0.084 & -0.046 & -0.075 & -0.052\\
## \hspace{1em} & (0.036) & (0.036) & (0.015) & (0.036) & (0.019)\\
## \hspace{1em}n & 8,104 & 8,612 & 8,104 & 8,104 & 8,104\\
## \addlinespace[0.5em]
## \multicolumn{6}{l}{\textit{Difference}}\\
## \midrule \hspace{1em}PSA - Control & 0.073 & 0.066 & 0.037 & 0.059 & 0.037\\
## \hspace{1em} & (0.063) & (0.062) & (0.030) & (0.063) & (0.031)\\
## \hspace{1em}Concerns - Control & 0.068 & 0.071 & 0.073* & 0.075 & 0.070+\\
## \hspace{1em} & (0.062) & (0.060) & (0.035) & (0.062) & (0.036)\\
## \hspace{1em}Concerns - PSA & 0.068 & 0.071 & 0.073* & 0.075 & 0.030\\
## \hspace{1em} & (0.062) & (0.060) & (0.035) & (0.062) & (0.038)\\
## \hspace{1em}Control mean & -0.056 & -0.049 & -0.047* & -0.057 & -0.047*\\
## \hspace{1em} & (0.045) & (0.044) & (0.020) & (0.045) & (0.023)\\
## \bottomrule
## \multicolumn{6}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01, *** p $<$ 0.001}\\
## \end{tabular}
## \begin{tablenotes}
## \item \footnotesize The sample is users in the evaluation stage. The response measure is the combine
## \end{tablenotes}
## \end{threeparttable}
```

```
## \end{table}
#####################################################
### code chunk number 30: heterogeneity_estimation_religion
#####################################################
#| eval = TRUE,
#| cache = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'


hetero_covariates_list<-c("religion_christian","religion_muslim")

## response function (man/above median)
for (covariate in hetero_covariates_list){
  ## A) Difference-in-means estimate (i.e, treating missing data as random)
  est_DM <- paste0("est_DM_", covariate)
  est_formula <- paste0(est_DM, " <- estimatr::lm_robust(response ~ treatment_group, data = df_eval[whi
  eval(parse(text = est_formula))

  est_DM_alt <- paste0("est_DM_", covariate, "_alt")
  est_formula_alt <- paste0(est_DM_alt, " <- estimatr::lm_robust(response ~ treatment_group, data = df_
  eval(parse(text = est_formula_alt))

  ## B) Imputing pre-test as post-test
  est_IMP <- paste0("est_IMP_", covariate)
  est_formula_IMP <- paste0(est_IMP, " <- estimatr::lm_robust(response_imputed ~ treatment_group, data =
  eval(parse(text = est_formula_IMP))

  est_IMP_alt <- paste0("est_IMP_", covariate, "_alt")
  est_formula_IMP_alt <- paste0(est_IMP_alt, " <- estimatr::lm_robust(response_imputed ~ treatment_grou
  eval(parse(text = est_formula_IMP_alt))

  ## C) Generalized random forest
  est_GRF <- paste0("est_GRF_", covariate)
  est_formula_GRF <- paste0(est_GRF, " <- mcf_estimate(df_eval[which(df_eval$", covariate, " == 1),], '
  eval(parse(text = est_formula_GRF))

  est_GRF_alt <- paste0("est_GRF_", covariate, "_alt")
  est_formula_GRF_alt <- paste0(est_GRF_alt, " <- mcf_estimate(df_eval_alt[which(df_eval_alt$", covariat
  eval(parse(text = est_formula_GRF_alt))

  ## D) IP Weighting
  est_IP <- paste0("est_IP_", covariate)
  est_formula_IP <- paste0(est_IP, " <- IP_estimate(df_eval[which(df_eval$", covariate, " == 1),], covar
  eval(parse(text = est_formula_IP))

  est_IP_alt <- paste0("est_IP_", covariate, "_alt")
  est_formula_IP_alt <- paste0(est_IP_alt, " <- IP_estimate(df_eval_alt[which(df_eval_alt$", covariate,
  eval(parse(text = est_formula_IP_alt))

  ## E) Lin estimator
  est_Lin <- paste0("est_Lin_", covariate)
```

```r
  if (covariate %in% c("religion_christian","religion_muslim")){
    covariate_lin_temp <- as.formula(paste('~', paste0(
      c('willingness_imputed', 'get_vaccinated_imputed'),
      collapse = ' + ')))
  }else {
    covariate_lin_temp<-covariate_lin
  }

  est_formula_Lin <- paste0(est_Lin, " <- estimatr::lm_lin(response ~ treatment_group, data = df_eval[wh
  eval(parse(text = est_formula_Lin))

  est_Lin_alt <- paste0("est_Lin_", covariate, "_alt")
  est_formula_Lin_alt <- paste0(est_Lin_alt, " <- estimatr::lm_lin(response ~ treatment_group, data = di
  eval(parse(text = est_formula_Lin_alt))
}



##########################################################
### code chunk number 31: heterogeneity_results_religion
##########################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'


# Combine model objects
out_list_christian <- list(
  DM = est_DM_religion_christian,
  IMP = est_IMP_religion_christian,
  Lin = est_Lin_religion_christian,
  IP = est_IP_religion_christian
)

out_list_muslim <- list(
  DM = est_DM_religion_muslim,
  IMP = est_IMP_religion_muslim,
  Lin = est_Lin_religion_muslim,
  IP = est_IP_religion_muslim
)
#
out_list_diff_religion <- list(
  DM = compare_models(est_DM_religion_muslim, est_DM_religion_christian)$Difference,
  IMP = compare_models(est_IMP_religion_muslim, est_IMP_religion_christian)$Difference,
  Lin = compare_models(est_Lin_religion_muslim, est_Lin_religion_christian)$Difference,
  IP = compare_models(est_IP_religion_muslim, est_IP_religion_christian)$Difference
)

out_list_diff_religion_alt <- list(
  DM = compare_models(est_DM_religion_christian_alt, est_DM_religion_muslim_alt)$Difference,
  IMP = compare_models(est_IMP_religion_christian_alt, est_IMP_religion_muslim_alt)$Difference,
  Lin = compare_models(est_Lin_religion_christian_alt, est_Lin_religion_muslim_alt)$Difference,
```

```
  IP = compare_models(est_IP_religion_christian_alt, est_IP_religion_muslim_alt)$Difference
)

#
out_list_christian <- modelsummary(out_list_christian,
                                   output = 'modelsummary_list')
out_list_muslim<- modelsummary(out_list_muslim,
                               output = 'modelsummary_list')
#
for(x in names(out_list_christian)){
  # add on Concerns - PSA coefficients
  out_list_christian[[x]]$tidy <- rbind(out_list_christian[[x]]$tidy |>
                                          select(term, estimate, std.error, statistic,
                                                 p.value),
                                        broom::tidy(get(paste0('est_', x, '_religion_christian_alt')))[3
                                          mutate(term = 'treatment_groupConcerns2') |>
                                          select(term, estimate, std.error, statistic,
                                                 p.value))

  out_list_muslim[[x]]$tidy <- rbind(out_list_muslim[[x]]$tidy |>
                                       select(term, estimate, std.error, statistic,
                                              p.value),
                                     broom::tidy(get(paste0('est_', x, '_religion_muslim_alt')))[3,] |>
                                       mutate(term = 'treatment_groupConcerns2') |>
                                       select(term, estimate, std.error, statistic,
                                              p.value))

  out_list_diff_religion[[x]]$tidy <- rbind(out_list_diff_religion[[x]]$tidy |>
                                              select(term, estimate, std.error, statistic,
                                                     p.value),
                                            out_list_diff_religion_alt[[x]]$tidy[
                                              which(out_list_diff_religion_alt[[x]]$tidy$term == 'treatr
                                              mutate(term = 'treatment_groupConcerns2') |>
                                              select(term, estimate, std.error, statistic,
                                                     p.value))

  # drop p-values for control condition so no stars
  out_list_christian[[x]]$tidy[which(out_list_christian[[x]]$tidy$term == '(Intercept)'),
                               c('p.value', 'statistic')] <- NA
  out_list_muslim[[x]]$tidy[which(out_list_muslim[[x]]$tidy$term == '(Intercept)'),
                            c('p.value', 'statistic')] <- NA
}

#
# # Add GRF; different format
out_list_christian[['GRF']] <- format_grf_summary(est_GRF_religion_christian)
out_list_christian[['GRF']]$tidy <- rbind(out_list_christian[['GRF']]$tidy,
                                          format_grf_summary(est_GRF_religion_christian_alt)$tidy[3,] |>
                                            mutate(term = 'treatment_groupConcerns2'))
out_list_christian[['GRF']]$tidy[which(out_list_christian[['GRF']]$tidy$term == '(Intercept)'),
                                 c('p.value', 'statistic')] <- NA

out_list_muslim[['GRF']] <- format_grf_summary(est_GRF_religion_muslim)
```

```r
out_list_muslim[['GRF']]$tidy <- rbind(out_list_muslim[['GRF']]$tidy,
                                       format_grf_summary(est_GRF_religion_muslim_alt)$tidy[3,] |>
                                         mutate(term = 'treatment_groupConcerns2'))
out_list_muslim[['GRF']]$tidy[which(out_list_muslim[['GRF']]$tidy$term == '(Intercept)'),
                              c('p.value', 'statistic')] <- NA

out_list_diff_religion[['GRF']] <- compare_models(format_grf_summary(est_GRF_religion_muslim),
                                                  format_grf_summary(est_GRF_religion_christian))$Diffe
out_list_diff_religion[['GRF']]$tidy <- rbind(out_list_diff_religion[['GRF']]$tidy,
                                              compare_models(format_grf_summary(est_GRF_religion_muslim
                                                             format_grf_summary(est_GRF_religion_christ
                                                filter(term == 'treatment_groupConcerns') |>
                                                mutate(term = 'treatment_groupConcerns2'))


modelsummary(list(`Religion: Christian` = out_list_christian,
                  `Religion: Muslim` = out_list_muslim,
                  Difference = out_list_diff_religion),
             shape= 'rbind',
             output = 'latex',
             coef_map = c('treatment_groupPSA' = 'PSA - Control',
                          'treatment_groupConcerns' = 'Concerns - Control',
                          'treatment_groupConcerns2' = 'Concerns - PSA',
                          '(Intercept)' = 'Control mean'),
             stars = TRUE ,
             gof_map = list(list('raw' = 'nobs',
                                 'clean' = 'n',
                                 'fmt' = f1)),
             escape = FALSE,
             caption= 'Treatment effect estimates and response under alternative approaches to estimati
  kable_styling(latex_options = c('HOLD_position')) |>
  footnote(general = paste0('\\\\footnotesize The sample is users in the evaluation stage. The response
  ),
  escape = FALSE,
  threeparttable = TRUE,
  general_title = '')
```

```
## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.

## Warning: The `caption` argument is not supported by `modelsummary`. Try `title` instead.
```

Table 10: Treatment effect estimates and response under alternative approaches to estimation.

	DM	IMP	Lin	IP	GRF
Religion: Christian					
PSA - Control	0.102**	0.090**	0.049**	0.082*	0.056***
	(0.035)	(0.034)	(0.017)	(0.035)	(0.017)
Concerns - Control	0.361***	0.323***	0.255***	0.320***	0.265***
	(0.035)	(0.034)	(0.020)	(0.035)	(0.020)
Concerns - PSA	0.259***	0.234***	0.206***	0.234***	0.205***
	(0.034)	(0.033)	(0.021)	(0.035)	(0.021)
Control mean	-0.211	-0.202	-0.155	-0.194	-0.159
	(0.025)	(0.025)	(0.011)	(0.025)	(0.013)
n	16,160	17,203	16,160	16,160	16,160
Religion: Muslim					
PSA - Control	-0.075	-0.079	-0.047	-0.072	-0.025
	(0.061)	(0.060)	(0.037)	(0.061)	(0.037)
Concerns - Control	0.174**	0.157**	0.099*	0.152*	0.119**
	(0.059)	(0.058)	(0.039)	(0.059)	(0.039)
Concerns - PSA	0.249***	0.236***	0.146***	0.229***	0.145***
	(0.059)	(0.058)	(0.041)	(0.060)	(0.041)
Control mean	0.295	0.298	0.314	0.302	0.298
	(0.043)	(0.042)	(0.025)	(0.043)	(0.019)
n	3,878	4,034	3,878	3,878	3,878
Difference					
PSA - Control	0.177*	0.169*	0.096*	0.154*	0.081*
	(0.070)	(0.069)	(0.040)	(0.070)	(0.041)
Concerns - Control	0.187**	0.166*	0.156***	0.168*	0.145***
	(0.068)	(0.067)	(0.044)	(0.069)	(0.044)
Concerns - PSA	-0.010	0.003	-0.060	-0.005	0.060
	(0.069)	(0.067)	(0.046)	(0.069)	(0.046)
Control mean	-0.505***	-0.500***	-0.469***	-0.496***	-0.457***
	(0.049)	(0.049)	(0.027)	(0.050)	(0.024)

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

The sample is users in the evaluation stage. The response measure is the combined index. Estimates are average treatment effects, and control mean. Estimating procedures are discussed in the text. Statistical significance is reported only for treatment effect estimates, and differences, not for baseline control means.

```
##################################################
### code chunk number 32: cost
##################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| strip.white=TRUE,
#| results='asis'
total_ad <- 8824.20
eval_ad <- 8095.75
nigeria_ad <- 3137.8
kenya_ad <- 4957.95
```

```r
concern_n <- sum(df_eval$treatment_group == 'Concerns')
concern_nigeria_n <- sum(df_nigeria$treatment_group == 'Concerns')
concern_kenya_n <- sum(df_kenya$treatment_group == 'Concerns')

concern_ad <- eval_ad * concern_n/eval_n
concern_ad_nigeria <- nigeria_ad * concern_nigeria_n/nigeria_n
concern_ad_kenya <- kenya_ad * concern_kenya_n/kenya_n

concern_ad_inc <- concern_ad + 0.75*concern_n
concern_ad_inc_nigeria <- concern_ad_nigeria + 0.75*concern_nigeria_n
concern_ad_inc_kenya <- concern_ad_kenya + 0.75*concern_kenya_n

eval_all <- eval_ad + 0.75*eval_n
pp_all <- eval_ad/eval_n + 0.75# total pp cost (ads + incentives)



##################################################
### code chunk number 33: cost_table
##################################################
#| eval = TRUE,
#| echo=FALSE,
#| warning=FALSE,
#| message=FALSE,
#| strip.white=TRUE,
#| results='asis'
df_eval <- df_eval |>
  mutate(get_vaccinated_bin = 1 * (get_vaccinated_2>2),
         willingness_bin = 1 * (willingness_2>2),
         either_bin = pmax(get_vaccinated_bin, willingness_bin, na.rm = TRUE))

df_kenya <- df_kenya |>
  mutate(get_vaccinated_bin = 1 * (get_vaccinated_2>2),
         willingness_bin = 1 * (willingness_2>2),
         either_bin = pmax(get_vaccinated_bin, willingness_bin, na.rm = TRUE))

df_nigeria <- df_nigeria |>
  mutate(get_vaccinated_bin = 1 * (get_vaccinated_2>2),
         willingness_bin = 1 * (willingness_2>2),
         either_bin = pmax(get_vaccinated_bin, willingness_bin, na.rm = TRUE))

est_bin_intention <- lm_lin(get_vaccinated_bin ~ treatment_group, data = df_eval,
                            covariates = covariate_lin)

est_bin_willingness <- lm_lin(willingness_bin ~ treatment_group, data = df_eval,
                              covariates = covariate_lin)

est_bin_either <-  lm_lin(either_bin ~ treatment_group, data = df_eval,
                          covariates = covariate_lin)

est_bin_intention_kenya <- lm_lin(get_vaccinated_bin ~ treatment_group,
                                  data = df_kenya,
                                  covariates = covariate_lin)
```

```r
est_bin_intention_nigeria <- lm_lin(get_vaccinated_bin ~ treatment_group,
                                    data = df_nigeria,
                                    covariates = covariate_lin)

est_bin_willingness_kenya <- lm_lin(willingness_bin ~ treatment_group,
                                    data = df_kenya,
                                    covariates = covariate_lin)
est_bin_willingness_nigeria <- lm_lin(willingness_bin ~ treatment_group,
                                      data = df_nigeria,
                                      covariates = covariate_lin)

est_bin_either_kenya <-  lm_lin(either_bin ~ treatment_group, data = df_kenya,
                                covariates = covariate_lin)

est_bin_either_nigeria <-  lm_lin(either_bin ~ treatment_group, data = df_nigeria,
                                  covariates = covariate_lin)

out_list <- modelsummary(list(Intention = est_bin_intention,
                              Willingness = est_bin_willingness,
                              Either = est_bin_either),
                         output = 'modelsummary_list')
out_list_kenya <- modelsummary(list(Intention = est_bin_intention_kenya,
                                    Willingness = est_bin_willingness_kenya,
                                    Either = est_bin_either_kenya),
                               output = 'modelsummary_list')
out_list_nigeria <- modelsummary(list(Intention = est_bin_intention_nigeria,
                                      Willingness = est_bin_willingness_nigeria,
                                      Either = est_bin_either_nigeria),
                                 output = 'modelsummary_list')

for(x in names(out_list)){
  # drop p-values for control condition so no stars
  out_list[[x]]$tidy[which(out_list[[x]]$tidy$term == '(Intercept)'),
                     c('p.value', 'statistic')] <- NA
  out_list_kenya[[x]]$tidy[which(out_list_kenya[[x]]$tidy$term == '(Intercept)'),
                           c('p.value', 'statistic')] <- NA
  out_list_nigeria[[x]]$tidy[which(out_list_nigeria[[x]]$tidy$term == '(Intercept)'),
                             c('p.value', 'statistic')] <- NA

  # Number of people influenced
  out_list[[x]]$glance['influenced'] <-
    out_list[[x]]$tidy[
      which(out_list[[x]]$tidy$term == 'treatment_groupConcerns'),
      'estimate'] *
    sum(df_eval$treatment_group == 'Concerns')

  out_list_kenya[[x]]$glance['influenced'] <-
    out_list_kenya[[x]]$tidy[
      which(out_list_kenya[[x]]$tidy$term == 'treatment_groupConcerns'),
      'estimate'] *
    sum(df_kenya$treatment_group == 'Concerns')

  out_list_nigeria[[x]]$glance['influenced'] <-
```

```
    out_list_nigeria[[x]]$tidy[
      which(out_list_nigeria[[x]]$tidy$term == 'treatment_groupConcerns'),
      'estimate'] *
    sum(df_nigeria$treatment_group == 'Concerns')

  # Cost per influenced person
  out_list[[x]]$glance['cost'] <- concern_ad/out_list[[x]]$glance['influenced']

  out_list_kenya[[x]]$glance['cost'] <- concern_ad_kenya/out_list_kenya[[x]]$glance['influenced']

  out_list_nigeria[[x]]$glance['cost'] <-  concern_ad_nigeria/out_list_nigeria[[x]]$glance['influenced']

  # Cost per influenced person including incentives
  out_list[[x]]$glance['cost_inc'] <- concern_ad_inc/out_list[[x]]$glance['influenced']

  out_list_kenya[[x]]$glance['cost_inc'] <- concern_ad_inc_kenya/out_list_kenya[[x]]$glance['influenced

  out_list_nigeria[[x]]$glance['cost_inc'] <-  concern_ad_inc_nigeria/out_list_nigeria[[x]]$glance['infl
}


modelsummary(list(Combined = out_list,
                  Kenya = out_list_kenya,
                  Nigeria = out_list_nigeria),
             shape = 'rbind',
             output = 'latex',
             coef_map = c('treatment_groupConcerns' = 'Concerns - Control',
                          '(Intercept)' = 'Control mean'),
             stars = TRUE ,
             gof_map = list(
               list('raw' = 'cost',
                    'clean' = 'Ad cost per influenced person',
                    'fmt' = fd),
               list('raw' = 'cost_inc',
                    'clean' = 'Total cost per influenced person',
                    'fmt' = fd)),
             escape = FALSE,
             title= 'Estimates on cost per influenced person.') |>
  kable_styling(latex_options = c('HOLD_position')) |>
  footnote(general = paste0('\\\\footnotesize The sample is users in the evaluation stage, $n = $ ',
                            prettyNum(eval_n, big.mark = ','),' overall, $n = $ ',
                            prettyNum(kenya_n, big.mark = ','),' in Kenya, and $n = $ ',
                            prettyNum(nigeria_n, big.mark = ','),' in Nigeria. The response measures ar
  ),
  escape = FALSE,
  threeparttable = TRUE,
  general_title = '')
```

Table 11: Estimates on cost per influenced person.

	Intention	Willingness	Either
Combined			
Concerns - Control	0.053***	0.045***	0.047***
	(0.004)	(0.006)	(0.004)
Control mean	0.787	0.547	0.810
	(0.003)	(0.004)	(0.003)
Ad cost per influenced person	$6.89	$8.15	$7.88
Total cost per influenced person	$20.97	$24.81	$23.98
Kenya			
Concerns - Control	0.067***	0.037***	0.056***
	(0.007)	(0.010)	(0.007)
Control mean	0.779	0.605	0.804
	(0.004)	(0.006)	(0.004)
Ad cost per influenced person	$10.21	$18.59	$12.26
Total cost per influenced person	$21.39	$38.94	$25.67
Nigeria			
Concerns - Control	0.046***	0.048***	0.042***
	(0.005)	(0.007)	(0.005)
Control mean	0.791	0.520	0.812
	(0.003)	(0.005)	(0.003)
Ad cost per influenced person	$4.58	$4.40	$5.06
Total cost per influenced person	$20.78	$19.96	$22.97

+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001

The sample is users in the evaluation stage, n = 22,052 overall, n = 7,235 in Kenya, and n = 14,817 in Nigeria. The response measures are binary transformation of the outcome for the *Intention* and *Willingness* measures, and for the *Either* measure, a variable that takes the value one if either of the other binary transformations is a one, and zero otherwise. Estimates are average treatment effects, and control means. Estimates are produced from a linear estimator, controlling for pre-test response (Lin, 2013). Statistical significance is reported only for treatment effect estimates, not for control means.

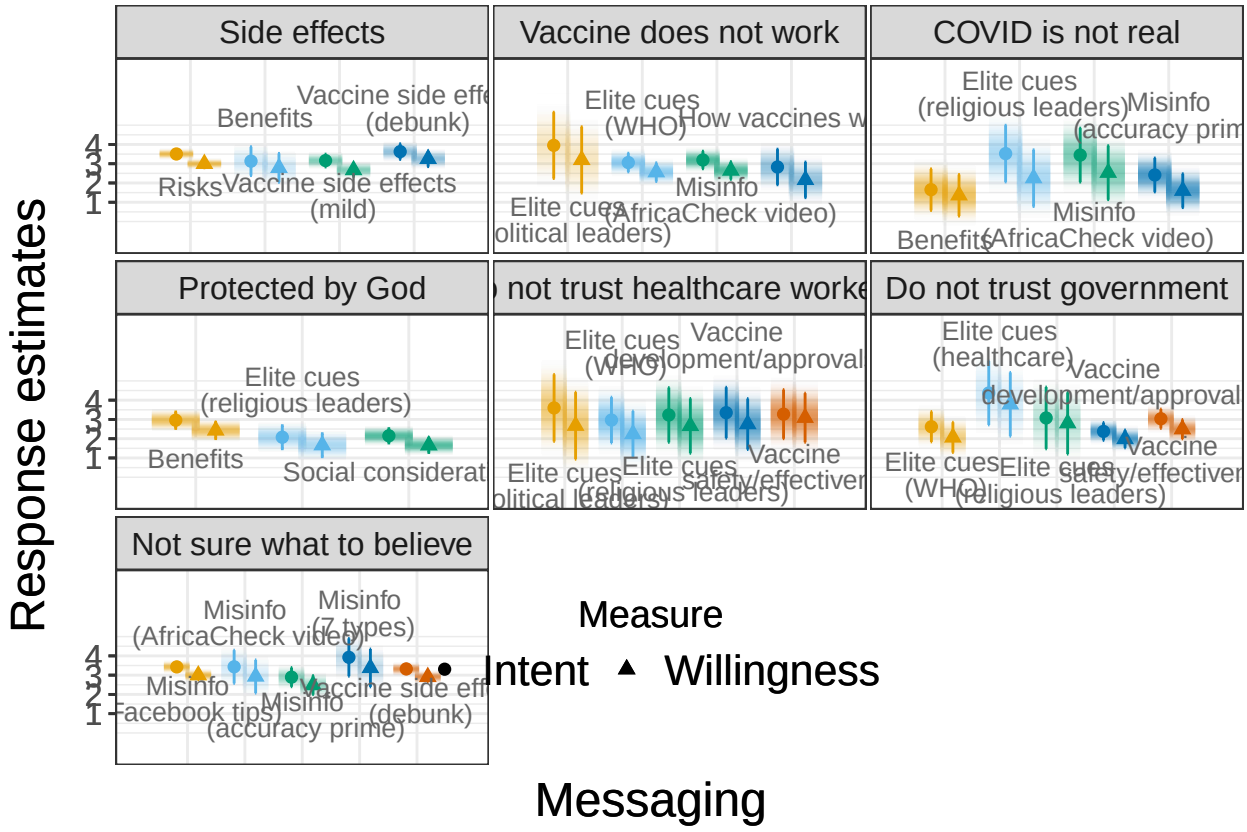
```
min_cost <- min(unlist(lapply(lapply(out_list, `[[`, 2), `[[`, 'cost')))
max_cost <- max(unlist(lapply(lapply(out_list, `[[`, 2), `[[`, 'cost')))

min_total_cost <- min(unlist(lapply(lapply(out_list, `[[`, 2), `[[`, 'cost_inc')))
max_total_cost <- max(unlist(lapply(lapply(out_list, `[[`, 2), `[[`, 'cost_inc')))


#####################################################
### code chunk number 34: learning_outcomes
#####################################################
#| eval = TRUE,
#| echo=FALSE, warning=FALSE, message=FALSE,
#| fig.align='center',
#| fig.width=12, fig.height=8,
#| prefix.string = 'fig',
#| fig.cap=
#| paste0('\\textbf{Vaccine intention and willingness response estimates in the
```

```
#|  learning stage.} ',
#| 'We estimate response separately across concern categories. ',
#| 'The sample is users who expressed only one concern in the learning stage,
#|  $n = $ ',
#| prettyNum(sum(! (duplicated(df_concerns$user_id) |
#| duplicated(df_concerns$user_id, fromLast=TRUE))), big.mark = ','),
#| '. We only consider these users in this analysis, to ensure there is no
#|  spillover across messaging. Estimates are of mean response, produced from
#|  the non-contextual stabilized variance weighting scheme discussed in
#|  Hadad et al. 2013. Error bars represent symmetric 95\\%
#|  confidence intervals.'),
#| strip.white=TRUE,
#| results='asis'
out <- list()
df_concerns1 <- df_concerns[! (duplicated(df_concerns$user_id) |
                                 duplicated(df_concerns$user_id, fromLast=TRUE)) ,]
for(cid in sort(unique(df_concerns$concern_id)) ){
  ddfh <- df_concerns1[which(df_concerns1$concern_id == cid),]

  df_probs <- as.matrix(ddfh[,paste0('prob_c', cid, '_m', sort(unique(ddfh$treatment_id) ))])

  balwts <- calculate_balwts(as.numeric(as.factor(ddfh$treatment_id)), df_probs)

  A <- nrow(ddfh)
  K <- length(unique(as.factor(ddfh$treatment_id)))

  aipw_scores_g <- banditsCI::aw_scores(
    ws = as.numeric(as.factor(ddfh$treatment_id)),
    yobs = ddfh$get_vaccinated_2_imputed,
    K = length(unique(as.factor(ddfh$treatment_id))),
    balwts = balwts
  )
  aipw_scores_w <- aw_scores(
    ws = as.numeric(as.factor(ddfh$treatment_id)),
    yobs = ddfh$willingness_2_imputed,
    K = length(unique(as.factor(ddfh$treatment_id))),
    balwts = balwts
  )

  policy1 <- lapply(1:K, function(x) {
    pol_mat <- matrix(0, nrow = A, ncol = K)
    pol_mat[,x] <- 1
    pol_mat
  }
  )

  df_probs <- df_probs/rowSums(df_probs)

  out_full_g <- banditsCI::output_estimates(
    policy1 = policy1,
    gammahat = aipw_scores_g,
    probs_array = df_probs,
    non_contextual_minvar = FALSE,
```

```
  non_contextual_stablevar = TRUE,
  non_contextual_twopoint = FALSE,
  contextual_minvar = FALSE,
  contextual_stablevar = FALSE,
  uniform = FALSE)

out_full_w <- banditsCI::output_estimates(
  policy1 = policy1,
  gammahat = aipw_scores_w,
  probs_array = df_probs,
  non_contextual_minvar = FALSE,
  non_contextual_stablevar = TRUE,
  non_contextual_twopoint = FALSE,
  contextual_minvar = FALSE,
  contextual_stablevar = FALSE,
  uniform = FALSE)


# Combine the data frames into a single data frame
combinedMatrix_g <- do.call(rbind,
                            lapply(out_full_g, `[`, 'non_contextual_stablevar',))
combinedMatrix_w <- do.call(rbind,
                            lapply(out_full_w, `[`, 'non_contextual_stablevar',))

gg_data <- data.frame(
  estimate = c(combinedMatrix_g[,'estimate'], combinedMatrix_w[,'estimate']),
  std.error = c(combinedMatrix_w[,'std.error'], combinedMatrix_w[,'std.error']),
  measure = factor(rep(c('Intent', 'Willingness'), each = K)),
  treatment_id = factor(rep(sort(unique(ddfh$treatment_id)), times = 2)),
  concern_id = cid)

gg_labels <- gg_data |>
  group_by(treatment_id) |>
  summarize(min.est = min(estimate),
            max.est = max(estimate),
            estimate = mean(estimate),
            std.error = max(std.error),
            concern_id = mean(concern_id),
            measure = 'Intent') |>
  mutate(estimate = ((seq_along(estimate))%%2) * (-2.5*std.error + min.est-0.9) + ((seq_along(estimat
  ungroup() |>
  mutate(label = gsub('\n', '\n\ ', messages[sort(unique(ddfh$treatment_id))]))

# Create the plot
out[[cid]] <- ggplot(gg_data,
                     aes(y = estimate,
                         x = treatment_id,
                         color = treatment_id,
                         fill = treatment_id,
                         group = measure,
                         shape = measure)) +
  geom_point(aes(y = estimate),
             size = 2,
```

```
                position = position_dodge(width=0.75)) +
    geom_errorbar(aes(ymin = estimate - 1.96*std.error,
                      ymax = estimate + 1.96*std.error), width = 0.05,
                  position = position_dodge(width=0.75), show.legend=FALSE) +
    stat_gradientinterval(aes(x = treatment_id,
                              ydist = distributional::dist_normal(estimate, std.error),
                              fill = treatment_id, group = measure),
                          width = 1,
                          position = position_dodge(0.75),
                          linewidth = 0,
                          point_size = 0,
                          point_alpha = 0,
                          interval_alpha = 0,
                          show.legend = FALSE,
                          fill_type = 'segments') +
    geom_text(data = gg_labels,
              aes(label=label),
              size = 3.6, color = 'grey40',
              hjust = 0.5, lineheight = .8) +
    facet_grid(~concern_id,
               labeller = labeller(concern_id = concerns)) +
    scale_color_manual(labels = as_labeller(messages),
                       values = cbPalette[-c(1, 5)]) +
    scale_fill_manual(labels = as_labeller(messages),
                      values = cbPalette[-c(1, 5)]) +
    vcf_theme() +
    scale_y_continuous(breaks = 1:4) +
    scale_x_discrete(expand = c(.25,.25)) +
    theme(#legend.position = 'None',
      legend.background=element_blank(),
      legend.title.align=0.5, legend.box.just = 'center',
      legend.key = element_rect(fill = NA),
      legend.text=element_text(size=16),
      axis.title.x=element_blank(),
      axis.title.y=element_blank(),
      axis.text.x=element_blank(),
      axis.ticks.x=element_blank(),
      plot.background = element_rect(fill='transparent',
                                     color = 'transparent'),
      plot.margin = margin(0, -10, 0, 0, 'pt')) +
    coord_cartesian(ylim = c(-1.2,8)) +
    guides(colour = 'none', fill = 'none',
           shape = guide_legend(title='Measure',
                                title.position = 'top'))

  if(cid %in% c(2,3,5,6)){
    out[[cid]] <- out[[cid]] +
      theme(axis.text.y=element_text(colour = '#FFFFFF00'),
            axis.ticks.y=element_line(colour = '#FFFFFF00'))
  }
}
```

```
## Warning: The `legend.title.align` argument of `theme()` is deprecated as of ggplot2 3.5.0.
## i Please use theme(legend.title = element_text(hjust)) instead.
```

```
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
legend <- cowplot::get_plot_component(out[[1]], "guide-box", return_all = TRUE)[[3]]
out <- lapply(out, function(x) x + theme(legend.position = 'None'))
out[[8]] <- legend

bottom_text <- grid::textGrob('Messaging', gp = grid::gpar(fontsize = 18))
left_text <- grid::textGrob('Response estimates', gp = grid::gpar(fontsize = 18), rot = 90, vjust = -0.4

g <- gridExtra::grid.arrange(grobs = out,
                             left = left_text,
                             bottom = bottom_text,
                             vp=grid::viewport(width=0.95, height=0.95))

grid::grid.draw(g)
```