Data Replication Script for Democracy, Public Support, and Measurement Uncertainty

2023-06-01

This file generates the latent variable estimates of support for democracy employed in the article, "Democracy, Public Support, and Measurement Uncertainty." It then combines these estimates with data on the control variables to create the analysis datasets.

Data Replication Files

There are 14 data files in the `data_replication` directory:

Type	Filename
Input	mood_dem.csv
Intermediate	claassen_input_raw.rds
Intermediate	claassen_replication_input.rda
Intermediate	exp_claassen_input.rds
Intermediate	claassen_replication_theta.rds
Intermediate	exp_theta.rds
Input	cls_ajps_cntrl_list.rds
Output	correct_cls_ajps.rda
Input	cls_apsr_cntrl_list.rds
Output	correct_cls_apsr.rda
Input	exp_ajps_cntrl_list.rds
Output	expcor_cls_ajps.rda
Input	exp_apsr_cntrl_list.rds
Output	expcor_cls_apsr.rda

"Input" files are exogenous. "Intermediate" files are generated by input files (and other intermediate files) and are used to generate output files; all of these processes are included in this document. "Output" files are generated by input and intermediate files, and are used in the analyses scripted in the `dcpo_demsupport.Rmd` file (as is the `exp_claassen_input.rds` intermediate file).

Source Data for Latent Variable Estimates

This section generates the source data needed to estimate the latent variable. The first chunk starts from the original surveys; it generates the file `claassen_input_raw.rds`. Collecting all of the needed survey datasets is labor-intensive (see https://github.com/fsolt/DCPOtools), so by default this code chunk is not evaluated and the next chunk simply uses the `claassen_input_raw.rds` file included in the replication materials.

```
claassen_input_raw <- DCPOtools:::claassen_setup(
    vars = read_csv(here("data_replication",
                         "mood_dem.csv"),
                    col_types = "cccccc"),
    file = here::here("data_replication",
```

```
                      "claassen_input_raw.rds"),
    datapath = "../../data/dcpo_surveys")
```

The object `claassen_input_raw` contains many more observations than the dataset employed in Claassen (2020a,b), and that dataset misidentifies the survey year in ~8% the observations it does include. The next chunk reproduces the sample of countries and years included in those data and saves it as `claassen_replication_input.rds`. The expanded dataset labeled "more data" in our work is saved as `exp_claassen_input.rds`.

```
claassen_input_raw <- rio::import(here::here("data_replication",
                      "claassen_input_raw.rds"))

dem <- readRDS(here("data_replication", "exp_ajps_cntrl_list.rds")) %>%
    first() %>%
    transmute(country, year, dem = as.numeric(Regime_VD > 1))

claassen_input_raw$scared <- claassen_input_raw %>%
    pluck("lower") %>%
    left_join(dem, by = join_by(country, year)) %>%
    mutate(r = case_when(r == -1 & dem == 0 ~ 999, # code non-respondents in
                         TRUE ~ r)) %>%  # autocracies as supporting democracy
    select(-dem)

remove_bad_items <- function(x) {
    x %>%
    filter(!(str_detect(survey, "army_wvs") & # WVS obs identified as problematic by Claassen
                     ((country=="Albania" & year==1998) |
                          (country=="Indonesia" & (year==2001 | year==2006)) |
                          (country=="Iran" & year==2000) |
                          (country=="Pakistan" & (year==1997 | year==2001)) | # 1996 in Claassen
                          (country=="Vietnam" & year==2001)) |
                     (str_detect(item, "strong_wvs") &
                          ((country=="Egypt" & year==2012) |
                               (country=="Iran" & (year==2000 | year==2007)) | # 2005 in Claassen
                               (country=="India") |
                               (country=="Pakistan" & (year==1997 | year==2001)) | # 1996 in Claassen
                               (country=="Kyrgyzstan" & (year==2003 | year==2011)) |
                               (country=="Romania" & (year==1998 | year==2005 | year==2012)) |
                               (country=="Vietnam" & year==2001))) |
                     (country %in% c("Puerto Rico", "Northern Ireland",
                                     "SrpSka Republic", "Hong Kong SAR China"))))
}

claassen_input_raw1 <- map(claassen_input_raw, function(df) {
    df %>%
        remove_bad_items() %>%
        with_min_yrs(2)
})

exp_claassen_input <- format_claassen(claassen_input_raw1)

rio::export(exp_claassen_input, file = here::here("data_replication",
                                        "exp_claassen_input.rds"))
```

```r
dir.create("data", showWarnings = FALSE)
rio::export(exp_claassen_input, file = here::here("data",
                                          "exp_claassen_input.rds"))

# Note that before running `dataverse::get_file()` below, one should set their personal token and serve
# Sys.setenv("DATAVERSE_KEY" = "exampleToken")
# Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")
# These values can be set to persist across R sessions using `usethis::edit_r_environ()`

if (!file.exists(here("data_replication", "supdem raw survey marginals.tab"))) {
    tempfile <- dataverse::get_file_by_doi("doi:10.7910/DVN/HWLW0J/RA8IJC") # AJPS replication file

    writeBin(tempfile, here("data_replication",
                                  "supdem raw survey marginals.tab"))
}

supdem <- read_csv(here::here("data_replication", "supdem raw survey marginals.tab"),
                   col_types = "cdcddcdc")

supdem1 <- supdem %>%                                                  # 1390 obs
    janitor::clean_names() %>%
    mutate(old_country = country,
           country = countrycode::countrycode(old_country, "country.name", "country.name"),
           i_claassen0 = tolower(item),
           i_claassen = case_when(str_detect(i_claassen0, "strong_arb") ~ "strong_arb",
                                  str_detect(i_claassen0, "strong_lapop") ~ "strong_lapop",
                                  str_detect(i_claassen0, "strong_pew") ~ "strong_pew",
                                  TRUE ~ i_claassen0),
           year = if_else(c_abb == "INS" & year == 2010 & i_claassen0 == "strong_pew",
                          2011,
                          year)) %>%
    with_min_yrs(2)

supdem_cy <- supdem1 %>%                                       # 1376 obs
    select(country, year, project) %>%
    distinct()

claassen_replication_input <- map(claassen_input_raw, function(df) {
    claassen_input_cy <- df %>%           # 1868 obs
        mutate(p_dcpo = str_extract(survey, "^[a-z]+"),
               project = case_when(p_dcpo == "afrob" ~ "afb",
                                   p_dcpo == "amb" ~ "lapop",
                                   p_dcpo == "arabb" ~ "arb",
                                   p_dcpo == "asiab" ~ "asb",
                                   p_dcpo == "asianb" ~ "asnb",
                                   p_dcpo == "neb" ~ "ndb",
                                   p_dcpo == "sasianb" ~ "sab",
                                   TRUE ~ p_dcpo),
               y_dcpo = year) %>%
        select(country, year, y_dcpo, survey, project) %>%
        unique()

    no_problems <- inner_join(supdem_cy, claassen_input_cy,
```

```
                          by = join_by(country, year, project))     # 1283 obs

needed <- anti_join(supdem_cy, claassen_input_cy,
                    by = join_by(country, year, project))           # 93 obs

available <- anti_join(claassen_input_cy, supdem_cy,
                       by = join_by(country, year, project))        # 585 obs

year_fixes <- left_join(needed, available,
                        by = c("country", "project"),
                        relationship = "many-to-many") %>% # 89 obs
    mutate(diff = year.x - year.y) %>%
    group_by(country, project, year.x) %>%
    mutate(closest_to_claassen = min(abs(diff))) %>%
    ungroup() %>%
    group_by(country, project, year.y) %>%
    mutate(closest_to_dcpo = min(abs(diff))) %>%
    ungroup() %>%
    filter(closest_to_claassen == abs(diff) & closest_to_dcpo == abs(diff) & abs(diff) <= 3) %>%
    filter(!(country == "Egypt" & year.x == 2014 & survey == "afrob5")) # double match (it's really

cys_crosswalk <- year_fixes %>%
    select(country, y_dcpo, y_claassen = year.x, survey)

missing_cyps <- anti_join(needed, year_fixes,
                          by = c("country", "year" = "year.x", "project")) # 4 obs; listed in issue

cys_to_drop <- anti_join(available, year_fixes,
                         by = c("country",
                                "year" = "year.y",
                                "project")) %>% # 496 obs
    select(-y_dcpo)

cri <- df %>% #3705
    remove_bad_items() %>%
    mutate(p_dcpo = str_extract(survey, "^[a-z]+"),
           project = case_when(p_dcpo == "afrob" ~ "afb",
                               p_dcpo == "amb" ~ "lapop",
                               p_dcpo == "arabb" ~ "arb",
                               p_dcpo == "asiab" ~ "asb",
                               p_dcpo == "asianb" ~ "asnb",
                               p_dcpo == "neb" ~ "ndb",
                               p_dcpo == "sasianb" ~ "sab",
                               TRUE ~ p_dcpo),
           i_dcpo = item,
           i_claassen = paste(str_replace(item, "_.*", "_"), p_dcpo)) %>%
    left_join(cys_crosswalk, by = c("country", "year" = "y_dcpo", "survey")) %>%
    anti_join(cys_to_drop, by = c("country", "year", "survey")) %>% # surveys unused by Claassen
    mutate(year = if_else(!is.na(y_claassen), y_claassen, year)) %>% # use Claassen's year codings
    mutate(item = if_else(item == "strong_amb_1" & year == 2004, "strong_amb_2", item)) %>% # items
    with_min_yrs(2) %>%
    # DCPOtools::format_claassen() %>%
    # pluck("data") %>%
```

```
        mutate(p_dcpo = str_extract(survey, "^[a-z]+"),
               project = case_when(p_dcpo == "afrob" ~ "afb",
                                   p_dcpo == "amb" ~ "lapop",
                                   p_dcpo == "arabb" ~ "arb",
                                   p_dcpo == "asiab" ~ "asb",
                                   p_dcpo == "asianb" ~ "asnb",
                                   p_dcpo == "neb" ~ "ndb",
                                   p_dcpo == "sasianb" ~ "sab",
                                   TRUE ~ p_dcpo),
               i_dcpo = item,
               i_claassen0 = paste0(str_replace(item, "_.*", "_"), project),
               i_claassen = case_when(i_claassen0 == "army_afb" ~ "army_afrob",
                                      i_claassen0 == "strong_afb" ~ "strong_afrob",
                                      i_claassen0 == "threestate_afb" ~ "threestate_afrob",
                                      i_claassen0 == "party_afb" ~ "party_afrob",
                                      i_claassen0 == "election_sab" ~ "elec_sab",
                                      TRUE ~ i_claassen0)) %>%
        inner_join(supdem1,
                   by = join_by(country, year, i_claassen)) %>%
        select(country, year, item = item.x, r, n, survey) %>%
        format_claassen()

    return(cri)
})

# each element of claassen_replication_input0 is missing 11 country-year-item observations included in

rio::export(claassen_replication_input,
            file = here("data_replication",
                        "claassen_replication_input.rds"))
```

Estimating Democratic Support

Below, the inputs created above are used to generate the latent variable estimates of democratic support. This requires a working installation of `cmdstanr`. Generating each set of estimates takes 5 to 10 minutes to run on a M1 MacBook Pro.

Original sample

```
claassen_replication_input <- rio::import(here("data_replication", "claassen_replication_input.rds"))

if (!file.exists(here("data_replication", "supdem.stan.mod5.stan"))) {
    tempfile <- dataverse::get_file_by_doi("doi:10.7910/DVN/HWLW0J/MKBL9E") # AJPS replication file

    writeBin(tempfile, here("data_replication", "supdem.stan.mod5.stan"))
}

cm5 <- cmdstan_model(here("data_replication", "supdem.stan.mod5.stan"))

warmup <- 250; sampling = 75

plan(multisession, workers = 12)
```

```
future_options <- furrr_options(seed = 324)

claassen_replication_output <- future_map(claassen_replication_input,
                                          \(cri_df) {
                                              cm5$sample(
                                                  data = cri_df,
                                                  max_treedepth = 14,
                                                  adapt_delta = 0.99,
                                                  step_size = 0.005,
                                                  seed = 324,
                                                  chains = 3,
                                                  parallel_chains = 3,
                                                  iter_warmup = warmup,
                                                  iter_sampling = sampling,
                                                  refresh = warmup/25
                                              )
                                          },
                                          .options = future_options
)

dat <- claassen_replication_input[["mar"]]$data %>%
    mutate(kk = claassen_replication_input[["mar"]]$jj,
           tt = claassen_replication_input[["mar"]]$tt)

kcodes <- dat %>%
    group_by(country) %>%
    summarize(kk = first(kk) %>%
                        as.numeric())

tcodes <- dat %>%
    group_by(year) %>%
    summarize(tt = first(tt))

ktcodes <- dat %>%
    group_by(country) %>%
    summarize(first_yr = min(year),
                     last_yr = max(year))

claassen_replication_theta <-
    imap(claassen_replication_output,
        \(res, idx) {
            res$draws("theta",
                      format = "df") %>%
                pivot_longer(starts_with("theta"), values_to = "theta") %>%
                mutate(tt = as.numeric(gsub("theta\\[(\\d+),\\d+\\]",
                                            "\\1", name)),
                       kk = as.numeric(gsub("theta\\[\\d+,(\\d+)\\]",
                                            "\\1", name))) %>%
                left_join(kcodes,
                          by = "kk") %>%
                left_join(tcodes, by = "tt") %>%
                mutate(year = if_else(tt == 1,
                                      as.integer(year),
```

```
                                        as.integer(min(year, na.rm = TRUE)
                                                   + tt - 1)),
                       set = idx) %>%
                left_join(ktcodes, by = "country") %>%
                arrange(kk, tt) %>%
                group_by(kk, tt) %>%
                mutate(draw = case_when(idx == "mar" ~ 0,
                                        idx == "lower" ~ n(),
                                        idx == "upper" ~ 2*n(),
                                        idx == "scared" ~ 3*n())
                           + 1:n()) %>%
                ungroup() %>%
                select(country, year, theta, set, draw)
        }) %>%
    list_rbind() %>%
    group_split(draw)

saveRDS(claassen_replication_theta,
        file = here("data_replication", "claassen_replication_theta.rds"))
```

Expanded 'More Data' Sample

```
exp_claassen_input <- rio::import(here("data_replication", "exp_claassen_input.rds"))

cm5 <- cmdstan_model(here("data_replication", "supdem.stan.mod5.stan"))

warmup <- 250; sampling = 75

plan(multisession, workers = 12)
future_options <- furrr_options(seed = 324)

exp_output <- future_map(exp_claassen_input,
                         \(cri_df) {
                             cm5$sample(
                                 data = cri_df,
                                 max_treedepth = 14,
                                 adapt_delta = 0.99,
                                 step_size = 0.005,
                                 seed = 324,
                                 chains = 3,
                                 parallel_chains = 3,
                                 iter_warmup = warmup,
                                 iter_sampling = sampling,
                                 refresh = warmup/25
                             )
                         },
                         .options = future_options
)

dat <- exp_claassen_input[["mar"]]$data %>%
    mutate(kk = exp_claassen_input[["mar"]]$jj,
           tt = exp_claassen_input[["mar"]]$tt)
```

```
kcodes <- dat %>%
    group_by(country) %>%
    summarize(kk = first(kk) %>%
                        as.numeric())

tcodes <- dat %>%
    group_by(year) %>%
    summarize(tt = first(tt))

ktcodes <- dat %>%
    group_by(country) %>%
    summarize(first_yr = min(year),
                    last_yr = max(year))

exp_theta <-
    imap(exp_output,
        \(res, idx) {
            res$draws("theta",
                      format = "df") %>%
                pivot_longer(starts_with("theta"), values_to = "theta") %>%
                mutate(tt = as.numeric(gsub("theta\\[(\\d+),\\d+\\]",
                                            "\\1", name)),
                       kk = as.numeric(gsub("theta\\[\\d+,(\\d+)\\]",
                                            "\\1", name))) %>%
                left_join(kcodes,
                          by = "kk") %>%
                left_join(tcodes, by = "tt") %>%
                mutate(year = if_else(tt == 1,
                                      as.integer(year),
                                      as.integer(min(year, na.rm = TRUE)
                                                 + tt - 1)),
                       set = idx) %>%
                left_join(ktcodes, by = "country") %>%
                arrange(kk, tt) %>%
                group_by(kk, tt) %>%
                mutate(draw = case_when(idx == "mar" ~ 0,
                                        idx == "lower" ~ n(),
                                        idx == "upper" ~ 2*n(),
                                        idx == "scared" ~ 3*n())
                                + 1:n()) %>%
                ungroup() %>%
                select(country, year, theta, set, draw)
        }) %>%
    list_rbind() %>%
    group_split(draw)

saveRDS(exp_theta,
        file = here("data_replication", "exp_theta.rds"))
```

Merging Data

The output of the next four chunks are the four files used in the (1) 'uncertainty' AJPS replication, (2) 'uncertainty' APSR replication, (3) 'uncertainty & more data AJPS replication', and (4) 'uncertainty & more

data' APSR replication.

AJPS replication, Claassen sample

```
# If latent-variable estimates were not just generated, load them now
if (!exists("claassen_replication_theta")) {
    claassen_replication_theta <- readRDS(file = here("data_replication", "claassen_replication_theta.r
}

cls_ajps_cntrl_list <- readRDS(here("data_replication", "cls_ajps_cntrl_list.rds"))

correct_cls_ajps <- map2(cls_ajps_cntrl_list,
                         claassen_replication_theta,
                         \(controls, theta) {
                             controls %>%
                                 left_join(theta, by = c("country", "year")) %>%
                                 mutate(SupDem_trim = ifelse(year < firstyear,
                                                             NA,
                                                             theta),
                                        theta_dem_trim = case_when(
                                            is.na(SupDem_trim) ~ NA,
                                            Regime_VD > 1 & !is.na(SupDem_trim) ~ theta,
                                            TRUE ~ 0),
                                        theta_aut_trim = case_when(
                                            is.na(SupDem_trim) ~ NA,
                                            Regime_VD <= 1 & !is.na(SupDem_trim) ~ theta,
                                            TRUE ~ 0)
                                        ) %>%
                                 select(country, year, theta, contains("trim"),
                                        everything())
                         })

dir.create("data", showWarnings = FALSE)
save(correct_cls_ajps, file = here("data","correct_cls_ajps.rda"))
```

APSR replication, Claassen sample

```
# If latent-variable estimates were not just generated, load them now
if (!exists("claassen_replication_theta")) {
    claassen_replication_theta <- readRDS(file = here("data_replication", "claassen_replication_theta.r
}

cls_apsr_cntrl_list <- readRDS(here("data_replication", "cls_apsr_cntrl_list.rds"))

correct_cls_apsr <- map2(cls_apsr_cntrl_list,
                         claassen_replication_theta,
                         \(controls, theta) {
    controls %>%
        left_join(theta, by = c("year", "country"))  %>%
        mutate(SupDem_trim = ifelse(year < firstyear, NA, theta)) %>%
        select(country, year, firstyear, theta, SupDem_trim,
               contains("z"), everything())
})
```

```
save(correct_cls_apsr, file = here("data","correct_cls_apsr.rda"))
```

AJPS replication, expanded sample

```
# If latent-variable estimates were not just generated, load them now
if (!exists("exp_theta")) {
    exp_theta <- readRDS(file = here("data_replication", "exp_theta.rds"))
}

exp_ajps_cntrl_list <- readRDS(here("data_replication", "exp_ajps_cntrl_list.rds"))

expcor_cls_ajps <- map2(exp_ajps_cntrl_list,
                  exp_theta,
                  \(controls, theta) {
                      controls %>%
                          left_join(theta, by = c("country", "year")) %>%
                          mutate(SupDem_trim = ifelse(year < firstyear, NA, theta),
                                 theta_dem_trim = case_when(
                                     is.na(SupDem_trim) ~ NA,
                                     Regime_VD > 1 & !is.na(SupDem_trim) ~ theta,
                                     TRUE ~ 0),
                                 theta_aut_trim = case_when(
                                     is.na(SupDem_trim) ~ NA,
                                     Regime_VD <= 1 & !is.na(SupDem_trim) ~ theta,
                                     TRUE ~ 0)) %>%
                          select(country, year, theta, contains("trim"), everything())
                  })

save(expcor_cls_ajps, file = here("data",
                                  "expcor_cls_ajps.rda"))
```

APSR replication, expanded sample

```
# If latent-variable estimates were not just generated, load them now
if (!exists("exp_theta")) {
    exp_theta <- readRDS(file = here("data_replication", "exp_theta.rds"))
}

exp_apsr_cntrl_list <- readRDS(here("data_replication", "exp_apsr_cntrl_list.rds"))

expcor_cls_apsr <- map2(exp_apsr_cntrl_list,
                        claassen_replication_theta,
                        \(controls, theta) {
                            controls %>%
                                left_join(theta, by = c("year", "country"))  %>%
                                mutate(SupDem_trim = ifelse(year < firstyear,
                                                            NA,
                                                            theta)) %>%
                                select(country, year, firstyear, theta, SupDem_trim,
                                       contains("z"), everything())
                        })
```

```
save(expcor_cls_apsr, file = here("data",
                                  "expcor_cls_apsr.rda"))
```