Main data building

Part of the final project for AQMSS II

Stepan Polikanov   Vera Okisheva

```r
## Building data for the analysis
## Dependencies: raw_data_prep.R, data_raw folder

source(here::here("utilities", "check_packages.R"))

conflicts_prefer(dplyr::filter)
```

```r
  ## raw ep data
  ep_raw_clean <- read_rds(here("data", "data_built", "ep_raw_clean.rds"))

  ## exit poll 18-03-24 11:21 CET
  # ep <- read_excel("data/exitpoll_18-03-24_11-21CET.xlsx")
  ep <- read_excel(here("data", "data_raw", "exitpoll_results",
                        "exitpoll_18-03-24_20-29CET.xlsx"), sheet = 2)

  ## yandex search data
  yandex_weekly <- read_csv(here("data", "data_raw", "migration",
                                 "df_countries_weekly.csv"))

  ## Migration

    ### fsb
    fsb_mig <- read_excel(here("data", "data_raw", "country_level",
                               "off_migration.xlsx"))

    ### Bilateral migration dataset
    bilat_migration <- read_csv(here("data", "data_raw", "migration",
                                     "bilat_mig_sex.csv"))

    ### International migrant stock

      #### Main
      intstock_migration <- read_excel(
        here("data", "data_raw", "migration",
             "undesa_pd_2020_ims_stock_by_sex_destination_and_origin.xlsx"),
        sheet = 2, skip = 9, guess_max = 37051
        )

  ## Religion
  wrp <- read_excel(here("data", "data_raw", "country_level", "wrp.xlsx"))

  ## Qog - econ development and democracy
  qog <- read_csv(
    here("data", "data_raw", "country_level", "qog_std_ts_jan24.csv"),
    guess_max = 15564
    )

  ## ATOP
  atop <- read_csv(here("data", "data_raw", "country_level", "atop5_1dy.csv"),
                   guess_max = 136648)

  ## Trade
  trade <- read_xlsx(here("data", "data_raw", "country_level", "trade.xlsx"),
                     sheet = 2)

  ## Handcoded
  hc <- read_xlsx(here("data", "data_raw", "country_level", "handcoded.xlsx"),
                  na = "NA") |>
    mutate(countrycode_n = as.numeric(countrycode_n)) |>
    drop_na(countrycode_c, countrycode_n) |>
    distinct(countrycode_c, countrycode_n, .keep_all = T)

  ## Distance
  geodist <- read_xls(here("data", "data_raw", "country_level",
                           "dist_cepii.xls"), na = ".")

  ## official election results

    ### 98% of processed ballots
    # off_res <- read_tsv("data/results-uik-20240318T1503UTC.tsv")

    ### Final data dumo (100 processed ballots)
    off_res <- read_tsv(here("data", "data_raw", "official_results",
                             "results-uik-20240320T0352UTC.tsv"))

  ## Supplementary data

    ### Voting station to country dictionary
    uik_dict <- read_excel(here("data", "data_raw", "official_results",
                                "uik_dictionary.xlsx"), sheet = 2)
    countrynameru_dict <- read_excel(here("data", "data_raw",
                                          "official_results",
                                          "uik_dictionary.xlsx"),
                                     sheet = 3)
```

```r
## Data cleaning: new ep
ep_cn <- ep |>
fill(country, .direction = "down") |>
  separate(col = "country", into = c("countryname_ru", "countryname_en"),
           sep = " / ") |>
  separate(col = "city", into = c("city_ru", "city_en"),
           sep = " / ") |>
  mutate(countryname_en = if_else(countryname_en == "Czech",
                                  "Czechia", countryname_en),
         countrycode_n = countrycode(countryname_en,
                                     origin = "country.name",
                                     destination = "iso3n"),
         countrycode_c = countrycode(countrycode_n,
                                     origin = "iso3n",
                                     destination = "iso3c"),
         voting_station = as.character(voting_station),
         city_en = if_else(str_detect(city_ru, "№") == T,
                           paste(city_en, str_sub(city_ru, -1,
                                                  nchar(city_ru))),
                           city_en),
         voters_counted = as.numeric(
           str_replace(str_replace(voters_counted, "~", ""), ">", "")),
         ep = 1)
```

```
## Aggregate to voting station level
ep.raw_agg <- ep_raw_clean |>
  group_by(voting_station, countryname_en, countryname_ru, countrycode_c,
           countrycode_n, city_ru, city_en) |>
  summarise(vote.raw_putin = sum(vote == "Putin")/n(),
            vote.raw_davankov = sum(vote == "Davankov")/n(),
            vote.raw_spoiled = sum(vote == "Spoiled ballot")/n(),
            share.raw_female = sum(sex == "Female", na.rm = T)/n(),
            share.raw_othergender = sum(sex == "Other", na.rm = T)/n(),
            share.raw_age1824 = sum(age_bin == "18-24", na.rm = T)/n(),
            share.raw_age2444 = sum(age_bin == "24-44", na.rm = T)/n(),
            share.raw_age4564 = sum(age_bin == "45-64", na.rm = T)/n(),
            share.raw_age65 = sum(age_bin == "65+", na.rm = T)/n(),
            share.raw_tourist =
              sum(out_of_Russia_time == "Tourist (lives in Russia)",
                  na.rm = T)/n(),
            share.raw_afterfeb =
              sum(out_of_Russia_time %in% c("6 month - 2 years",
                                            "< 6 months",
                                            "< 2 years"), na.rm = T)/n(),
            share.raw_before2014 =
              sum(out_of_Russia_time == "> 10 years", na.rm = T)/n(),
            share.raw_after2014 =
              sum(out_of_Russia_time == "> 5 years", na.rm = T)/n(),
            share.raw_2019 =
              sum(out_of_Russia_time == "2 - 5 years", na.rm = T)/n(),
            # ref 2014
            share.raw_afterfebref =
              sum(out_of_Russia_time %in% c("6 month - 2 years",
                                            "< 6 months",
                                            "< 2 years"), na.rm = T)
            /sum(out_of_Russia_time == "> 10 years", na.rm = T),
            share.raw_after2014ref =
              sum(out_of_Russia_time == "> 5 years", na.rm = T)
            /sum(out_of_Russia_time == "> 10 years", na.rm = T),
            share.raw_2019ref =
              sum(out_of_Russia_time == "2 - 5 years", na.rm = T)
            /sum(out_of_Russia_time == "> 10 years", na.rm = T),
            share.raw_timemore4h =
              sum(time_to_vs == "> 4 hours (staying for the night)",
                  na.rm = T)/n(),
            share.raw_local =
              sum(time_to_vs %in% c("<30 minutes", "30 minutes - 1 hour"),
                  na.rm = T)/n(),
            share.raw_trustanyyes =
              sum(result_trust %in% c("Definitely yes", "Probably yes"),
                  na.rm = T)/n(),
            share.raw_trustanyno =
              sum(result_trust %in% c("Definitely no", "Probably no"),
                  na.rm = T)/n()) |>
  ungroup() |>
  select(-countryname_ru, -countryname_en, -city_ru, -city_en)
```

```
## Convert countrynames in Russian provided by the Russian Ministry of Foreign
## Affairs to match with those int the international code dictionary
uik_dict_clean <- uik_dict |>
  mutate(uik = as.character(uik), # For merges
         country_compatible = case_when(
           country == "  -                  " ~ "       ",
           country == " -          " ~ "  -  ",
           country == "                  (  )"
           ~ "                 ",
           country == "          " ~ "     ",
           country == "                  " ~ "    ",
           country == "            " ~ "   ",
           country == "    " ~ "                  ",
           country == "   " ~ "                ",
           country == "         " ~ "    ",
           country == "         " ~ "   ",
           country == "     " ~ "           ",
           country == "        ,         " ~ "        ",
           country == "     ,            " ~ "     ",
           country == "     " ~ "       (     )",
           country == "    " ~ "          ",
           country == "           " ~ "    ",
           country == "         " ~ "    ",
           country == "       " ~ "   ",
           country == "                 " ~ "      ",
           country == "  " ~ "         -              ",
           country == "  -’    " ~ "  - '  ",
           country == "  " ~ "  ",
           country == "    " ~ "      ",
           country == "                  " ~ "    ",
           country == "  " ~ "    (          )",
           country == "                " ~ "     ",
           country == "          " ~ "    ",
           country == "         " ~ "    ",
           country == "         -   " ~ "   ",
           country == "        " ~ "   ",
           country == "     ,                   "
           ~ "      (              )",
           country == "       "
           ~ "                                 ",
           country == "   " ~ "   -      ",
           country == "           " ~ "    ",
           country == "    " ~ "     (               )",
           country == "       " ~ "   ",
           country == "     " ~ "     ",
           country == "       -         " ~ "   ",
           country == "  " ~ "          -              ",
           .default = country)) |>
  left_join(countrynameru_dict,
            by = join_by("country_compatible" == "Country or Area"))
```

```
## Select international voting stations
off_res_select <- off_res |>
  separate(col = "uik", into = c(NA, "uik_num"), sep = "№") |>
  filter(nchar(uik_num) == 4 & substring(uik_num, 1, 1) == "8") |>
  left_join(select(ep_cn, countryname_en, countryname_ru, countrycode_c,
                   countrycode_n, city_ru, city_en, voting_station),
            by = join_by("uik_num" == "voting_station")) |>
  left_join(uik_dict_clean, by = join_by("uik_num" == "uik")) |>
```

```
mutate(countryname_ru = if_else(is.na(countryname_ru),
                                country_compatible, countryname_ru),
       city_en = if_else(is.na(city_en), settlement, city_en),
       countrycode_c = if_else(
         tik == "         (          )",
         "KAZ", countrycode_c),
       countryname_ru = if_else(
         tik == "         (          )",
         "     ", countryname_ru),
       city_en = if_else(tik == "         (           )",
                         "Baikonur", city_en),
       city_ru = if_else(tik == "         (           )",
                         "    ", city_ru),
       countrycode_c = if_else(is.na(countrycode_c), `ISO-alpha3 code`,
                               countrycode_c),
       countrycode_n = if_else(is.na(countrycode_n),
                               countrycode(sourcevar = countrycode_c,
                                           origin = "iso3c",
                                           destination = "iso3n"),
                               countrycode_n),
       countryname_en = if_else(is.na(countryname_en),
                                countrycode(sourcevar = countrycode_n,
                                            origin = "iso3n",
                                            destination = "country.name"),
                                countryname_en)) |>
select(voting_station = uik_num, city_ru, city_en,
       countryname_ru, countryname_en, countrycode_c, countrycode_n,
       voters_in_list = `         ,                   `,
       ballots_received = `                 ,                           `,
       early_ballots = `                 ,            ,              `,
       ballots_voting_space = `                 ,                                 `,
       ballots_out_of_voting_space = `                 ,                                  `,
       ballots_destroyed = `                       `,
       ballots_in_movable_boxes = `                                      `,
       ballots_in_stationary_boxes = `                                `,
       ballots_invalid = `                          `,
       ballots_valid = `                         `,
       ballots_lost = `                       `,
       ballots_uncounted = `                 ,               `,
       davankov.abs_full = `                `,
       davankov_full = `                (%)`,
       putin.abs_full = `               `,
       putin_full = `                (%)`,
       slutsky.abs_full = `              `,
       slutsky_full = `               (%)` ,
       haritonov.abs_full = `                `,
       haritonov_full = `                (%)`) |>
mutate(across(c("putin_full", "davankov_full",
                "slutsky_full", "haritonov_full"),
              ~ ./100),
       spoiled_full = ballots_invalid/(early_ballots + ballots_voting_space
                                       + ballots_out_of_voting_space
                                       + ballots_lost + ballots_uncounted),
       countryname_en = case_when(countryname_en == "USA" ~ "United States",
                                  countryname_en == "UAE" ~ "United Arab Emirates",
                                  .default = countryname_en),
       countryname_ru = case_when(countryname_en == "  " ~ "              ",
                                  countryname_en == "  " ~ "                ",
                                  .default = countryname_ru))
```

```
## Aggregating yandex data
  yandex_24feb <- yandex_weekly |>
    mutate(across(c("date_from", "date_to"), ~ ymd(.)),
           period = case_when(date_to > "2022-02-24"
                              & date_from < "2022-09-21" ~ "feb24_sep21",
                              date_to > "2022-09-21" ~ "after_sep21",
                              .default = "before_feb24")) |>
    group_by(destination_country, period) |>
    summarize(searches = mean(count)) |>
    pivot_wider(id_cols = "destination_country",
                names_from = "period", values_from = "searches") |>
    mutate(diff.feb24_sep21 = feb24_sep21 - before_feb24,
           diff.after_sep21 = after_sep21 - before_feb24)
```

```
  ## Isolate RUS as origin, select countries with an exit poll (not good)
  out_ru <- bilat_migration |>
    filter(dest %in% unique(off_res_select$countrycode_c),
           orig == "RUS", year0 %in% c(1990, 2000, 2005, 2010, 2015)) |>
    group_by(dest, year0) |>
    summarise(mig_rate = mean(mig_rate)) |>
    pivot_wider(id_cols = dest, names_from = year0,
                values_from = mig_rate, names_prefix = "mig_rate_")
```

```
## Isolate Russia as origin, select all countries where vote was held (good?)
  intstock_ru <- intstock_migration |>
    rename_with(~ c("all_1990", "all_1995", "all_2000", "all_2005", "all_2010",
                    "all_2015", "all_2020", "male_1990", "male_1995",
                    "male_2000", "male_2005", "male_2010", "male_2015",
                    "male_2020", "female_1990", "female_1995", "female_2000",
                    "female_2005", "female_2010", "female_2015", "female_2020"),
                8:28) |>
    filter(`Location code of origin` == "643",
           `Location code of destination` %in% off_res_select$countrycode_n) |>
    select(countrycode_n = `Location code of destination`, 8:28) |>
    mutate(diff2015_2010 = (all_2015 - all_2010)/all_2010,
           diff2020_2015 = (all_2020 - all_2010)/all_2015)
```

```
## Select variables
  off_migration <- fsb_mig |>
    filter(goals == "            ") |>
    mutate(countrycode_n =
             case_when(country == "     " ~ 36,
                       country == "    ,            " ~ 40,
                       country == "      ,            " ~ 31,
                       country == "    ,          " ~ 8,
                       country ==
                         "   ,                         " ~ 12,
                       country == "   ,          " ~ 24,
                       country == "     ,             " ~ 32,
                       country == "    ,          " ~ 51,
                       country ==
                         "     ^                        " ~ 4,
                       country == "    ,           " ~ 48,
                       country == "     ,                " ~ 50,
                       country == "   ,         " ~ 204,
                       country == "    ,           " ~ 112,
                       country == "    ,           " ~ 56,
country == "    ,            " ~ 100,
country ==
  "    ,           " ~ 68,
country == "            " ~ 70,
country == "    ,            " ~ 72,
country ==
  "    ,                   " ~ 76,
# No Burundi
country == "   -      " ~ 96,
country == "    ,            " ~ 348,
country ==
  "     ,                    " ~ 862,
country == "    ,                    " ~ 704,

country == "   ,            " ~ 266,
country == "   ,           " ~ 328,
country == "  ,          " ~ 288,
country == "     ,            " ~ 320,
country == "   ,             " ~ 324,
# No Guinea-Bissau
country == "    ,                   " ~ 276,
country ==
  "             ,        ,                      " ~ 275,
country == "   ,            " ~ 300,
country == "   ,          " ~ 208,
country ==
  "   ,               ,                   " ~ 180,
country == "    ,           " ~ 262,
country == "   ,                " ~ 818,
country == "   ,           " ~ 894,
country == "    ,            " ~ 716,
country == "    ,           " ~ 376,
country == "   ,          " ~ 356,
country == "     ,            " ~ 360,
country == "    ,                    " ~ 400,
country == "  ,          " ~ 368,
country ==
  "  ,            ,                " ~ 364,
country == "    " ~ 372,
country == "    ,            " ~ 352,
country == "    ,           " ~ 724,
country == "   ,             " ~ 380,
country == "  -   ,         -   " ~ 132,
country == "     ,            " ~ 398,
country == "    ,            " ~ 116,
country == "    ,           " ~ 120,

country == "   ,          " ~ 634,
country == "   ,          " ~ 404,
country == "  ,          " ~ 196,
country == "   ,                 " ~ 156,
country == "    ,            " ~ 170,
country == "   ,          " ~ 178,
country ==
  "   ,     -              ,          -               " ~ 408,
country == "   -  ,          -  " ~ 188,
country == "   `   ,          `   " ~ 384,
country == "  ,          " ~ 192,
country == "   ,           " ~ 414,
country == "    ,             " ~ 417,
country == "         –              " ~ 418,
country == "    ,            " ~ 428,
country == "   ,            " ~ 422,
country ==
"                 ,                               " ~ 434,
country == "   ,            " ~ 440,
country == "      ,                 " ~ 442,
country == "      ,                  " ~ 478,
country == "     ,           " ~ 480,
country == "      ,            " ~ 450,
country == "    " ~ 458,
country == "  ,          " ~ 466,
country == "    ,          " ~ 470,
country == "    ,           " ~ 504,
country == "    ,                  " ~ 484,
country == "     ,           " ~ 508,
country == "    ,      ,           " ~ 498,
country == "    " ~ 496,
country == "    ,       " ~ 104,
country == "    ,           " ~ 516,
country == "   ,          " ~ 524,
country == "    ,                  " ~ 566,
country == "      ,             " ~ 528,
country == "     ,            " ~ 558,
country == "        " ~ 554,
country == "     ,            " ~ 578,
country == "                " ~ 784,
country == "  ,         " ~ 512,
country == "     ,                 " ~ 586,
country == "    ,          " ~ 591,
# No Paraguay
country == "  ,          " ~ 604,
country == "    ,          " ~ 616,
country == "      ,              " ~ 620,
country == "   ,      ,          " ~ 410,
country == "    ,             " ~ 646,
country == "    " ~ 642,
country ==
  "          ,                 " ~ 682,
# No North Macedonia
country == "    ,           " ~ 690,
country == "    ,           " ~ 686,
country == "   " ~ 688,
country == "     ,           " ~ 702,
country == "                " ~ 760,
country == "     ,            " ~ 703,
country == "     ,           " ~ 705,
country == "          ,               " ~ 840,
country == "   ,          " ~ 729,
country ==
"     ,             ,                   " ~ 834,
country == "      ,             " ~ 762,
country == "    ,           " ~ 764,
country == "   ,            " ~ 788,
country == "     ,       " ~ 795,
country == "    ,           " ~ 792,
country == "    ,          " ~ 800,
country == "      ,            " ~ 860,
country == "    ,                 " ~ 858,
country == "     ,            " ~ 608,
                       country == "     ,             " ~ 246,
                       country == "    ,             " ~ 250,
                       country == "     ,           " ~ 191,
                       # No Cetral African Republic
                       country == "  ,         " ~ 148,
                       country == "      " ~ 499,
                       country == "           " ~ 203,
                       country == "   ,         " ~ 152,
                       country == "     ,              " ~ 756,
                       country == "    ,           " ~ 752,
                       country ==
                         "  -   ,                           -   " ~ 144,
                       country == "    ,           " ~ 218,
                       country == "    ,            " ~ 233,
                       country ==
                         "    ,                            " ~ 231,
                       country == "    " ~ 232,
                       country == "       ,    -            " ~ 710,
                       country == "    " ~ 388,
                       country == "    " ~ 392,
                       country == "    " ~ 124,
                       country ==
                         "             ,                                  " ~ 826,
                       .default = NA),
           countrycode_c = countrycode(countrycode_n, origin = "iso3n",
                                       destination = "iso3c"),
           across(c(3:58), ~ if_else(is.na(.), 0, .))) |>
    drop_na(countrycode_n) |>
    rename_with(.fn = ~ paste0(rep(1:4, 4), "_", rep(2010:2023, each = 4)),
                .cols = c(3:58)) |>
    pivot_longer(cols = 3:51, names_to = c("quart", "year"),
                 names_sep = "_", values_to = "tourist_trips") |>
    group_by(countrycode_n, countrycode_c) |>
    summarize(mean_trips = mean(tourist_trips))

# No Burundi, Central African Republic, Paraguay, Guinea-Bissau, North Macedonia
```

```
## Select variables, homogenize countrycodes
wrp_select <- wrp |>
  group_by(COUNTRY) |>
  filter(YEAR == 2010) |>
  ungroup() |>
  mutate(countrycode_n = case_when(is.na(NUMISO) | NUMISO == 0
                                   ~ countrycode(COWCODE, origin = "cown", destination = "iso3n"),
                                   .default = countrycode(NUMISO, origin = "iso3n", destination = "iso3n")
  ),
  countrycode_c = countrycode(countrycode_n, origin = "iso3n", destination = "iso3c")) |>
  select(countrycode_n, countrycode_c, orthodox_num = CHRSORTH,
         orthodox_share = CHORTPCT) |>
  drop_na(countrycode_n, countrycode_c)
```

```
## Select variables and years
qog_select <- qog |>
  select(ccode, year, cname, vdem_polyarchy, bmr_dem, mad_gdppc,
         wdi_gdpcapcon2015) |>
  filter(year >= 2018) |>
  pivot_wider(names_from = year,
              values_from = c(vdem_polyarchy, bmr_dem, mad_gdppc,
                              wdi_gdpcapcon2015)) |>
  select(ccode, cname, vdem_polyarchy_2022, bmr_dem_2020,
         mad_gdppc_2018, wdi_gdpcapcon2015_2022) |>
  mutate(countrycode_n = countrycode(ccode, origin = "iso3n",
                                     destination = "iso3n"),
         countrycode_c = countrycode(countrycode_n, origin = "iso3n",
                                     destination = "iso3c")) |>
  drop_na(countrycode_n, countrycode_c)
```

```
## Select variables
atop_select <- atop |>
  filter(year == 2018, mem1 == 365 | mem2 == 365) |>
  mutate(countrycow = case_when(mem1 == 365 ~ mem2,
                                mem2 == 365 ~ mem1),
         countrycode_n = countrycode(countrycow, origin = "cown",
                                     destination = "iso3n"),
         countrycode_c = countrycode(countrycode_n, origin = "iso3n",
                                     destination = "iso3c")) |>
  right_join(distinct(off_res_select, countrycode_n, countrycode_c),
             by = c("countrycode_n", "countrycode_c")) |>
  transmute(countrycode_n, countrycode_c,
            nonagg = case_when(is.na(nonagg) ~ 0, nonagg == 1 ~ 1),
            consul = case_when(is.na(consul) | consul == 0 ~ 0,
                               consul == 1 ~ 1),
            neutral = case_when(is.na(neutral) | neutral == 0 ~ 0,
                                neutral == 1 ~ 1),
            defense = case_when(is.na(defense) | defense == 0 ~ 0,
                                defense == 1 ~ 1),
            obl_type = factor(rowSums(pick(nonagg, consul, neutral, defense)),
                              levels = 0:4,
                              labels = c("0", "1", "2", "3", "4")))
```

```
## Homogenize countrycodes
trade_select <- trade |>
  transmute(countryname = case_when(`Partner Name` ==
                                   "Ethiopia(excludes Eritrea)" ~ "Ethiopia",
                                 `Partner Name` ==
                                   "United States Minor Outlying I" ~ NA,
                                 `Partner Name` ==
                                   "Serbia, FR(Serbia/Montenegro)" ~ NA,
                                 .default = `Partner Name`),
            countrycode_n = countrycode(countryname,
                                        origin = "country.name",
                                        destination = "iso3n"),
            countrycode_c = countrycode(countrycode_n, origin = "iso3n",
                                        destination = "iso3c"),
            export_value = `Export (US$ Thousand)`,
            import_value = `Import (US$ Thousand)`,
            export_share = `Export Partner Share (%)`,
            import_share = `Import Partner Share (%)`) |>
  drop_na(countrycode_n, countrycode_c)
```

```
## Homogenize countrycodes
  geodist_select <- geodist |>
    filter(iso_o == "RUS") |>
    mutate(countrycode_n = countrycode(iso_d, origin = "iso3c",
                                      destination = "iso3n"),
           countrycode_c = countrycode(countrycode_n, origin = "iso3n",
                                      destination = "iso3c")) |>
    drop_na(countrycode_n, countrycode_c) |>
    select(countrycode_n, countrycode_c, dist, distcap, distw, distwces)
```

```
## Compile the final dataset
data_built <- off_res_select |>
  drop_na(countrycode_c, countrycode_n) |>
  full_join(ep_cn, by = c("voting_station", "countrycode_c",
                          "countrycode_n",
                          "city_en", "city_ru")) |>
  left_join(ep.raw_agg, by = c("voting_station", "countrycode_c",
                               "countrycode_n")) |>
  left_join(out_ru, by = join_by("countrycode_c" == "dest")) |>
  left_join(intstock_ru, by = "countrycode_n") |>
  left_join(wrp_select, by = c("countrycode_c", "countrycode_n")) |>
  left_join(qog_select, by = c("countrycode_c", "countrycode_n")) |>
  left_join(atop_select, by = c("countrycode_c", "countrycode_n")) |>
  left_join(trade_select, by = c("countrycode_c", "countrycode_n")) |>
  left_join(hc, by = c("countrycode_n", "countrycode_c")) |>
  left_join(geodist_select, by = c("countrycode_c", "countrycode_n")) |>
  left_join(off_migration, by = c("countrycode_c", "countrycode_n")) |>
  mutate(putin_full = if_else(countrycode_c == "TUR", putin_cec, putin_full),
         davankov_full = if_else(countrycode_c == "TUR", davankov_cec, davankov_full),
         spoiled_full = if_else(countrycode_c == "TUR", spoiled_cec, spoiled_full),
         slutsky_full = if_else(countrycode_c == "TUR", slutskiy_cec, slutsky_full),
         haritonov_full = if_else(countrycode_c == "TUR", haritonov_cec, haritonov_full))
```

```
# Also remove some of the variables
data_country <- data_built |>
  drop_na(countrycode_n, countrycode_c) |>
  group_by(countrycode_n, countrycode_c) |>
  summarize(
    # Different voting metrics from official results
    # (Istanbul was damaged during parsing - confirmed by author)
    across(c(voters_in_list:ballots_uncounted), ~ sum(.)),
    # Official results - shares
    across(c(davankov_full, putin_full, slutsky_full,
             haritonov_full, spoiled_full), ~ mean(.)*100),
    # Official results - absolute values
    across(c(davankov.abs_full, putin.abs_full, slutsky.abs_full,
             haritonov.abs_full), ~ sum(.)),
    # Exit poll - metrics
    across(c(voters_surveyed, voters_counted,
             # Ballots in boxes replaces the variable for Istanbul
             ballots_in_boxes, removed_or_destroyed.abs), ~ sum(.)),
    # Exit poll - shares
    across(c(removed_or_destroyed, putin_ep, davankov_ep, slutskiy_ep,
             haritonov_ep, spoiled_ep), ~ mean(., na.rm = T)*100),
    # Exit poll - absolute values
    across(c(putin.abs_ep, davankov.abs_ep,
             slutskiy.abs_ep, haritonov.abs_ep), ~ sum(.)),
    # Raw variables - shares
    across(c(share.raw_female, share.raw_othergender,
             share.raw_age1824, share.raw_age2444,
             share.raw_age4564, share.raw_age65,
             share.raw_tourist, share.raw_afterfeb,
             share.raw_before2014, share.raw_after2014,
             share.raw_2019, share.raw_afterfebref,
             share.raw_after2014ref, share.raw_2019ref,
             share.raw_timemore4h, share.raw_local,
             share.raw_trustanyyes, share.raw_trustanyno), ~ mean(.)*100),
    # Country level variables
    across(c(mig_rate_1990:mean_trips),
           # Ensure that all variables are country-level
           ~ if_else(all(. == first(.)), first(.), NA))) |>
  mutate(ep = if_else(!is.na(putin_ep), 1, 0),
         friendly_status = relevel(
           factor(friendly_status, levels = 0:2,
                  labels = c("Unfriendly", "Neutral", "Friendly")),
           ref = "Neutral"),
         putin_dist = putin_full - putin_ep,
         davankov_dist = davankov_full - davankov_ep,
         spoiled_dist = spoiled_full - spoiled_ep)
```

```
## Save the final datasets
write_rds(data_built, here("data", "data_built", "data_built_.rds"))
write_rds(data_country, here("data", "data_built", "data_country.rds"))

## Save the raw dataset
write_rds(ep_raw_clean, here("data", "data_built", "ep_raw_clean.rds"))
```