###########################################################################

####READ ME!

#CoupCats - please skim to look for "####CT:..." each time you open this. I'm making notes as I clean things to help you learn some stuff. Once you read the note go ahead and delete it.

###########################################################################

rm(list=ls())

#setwd("C:/Users/Camila/OneDrive/R related/R/R") # Set working file. #Camila

setwd("C:/Users/clayt/OneDrive - University of Kentucky/elements/current\_research/coupcats") #Clay at home

#load packages: if you use new packages, try to remember to add them to the packages on github. Otherwise, just flag them and I'll do it.

#source("https://raw.githubusercontent.com/thynec/CoupCats/refs/heads/main/packages.R")

#load libraries: if you add new libraries, try to remember to add them to github. Otherwise, just flag them and I'll do it.

source("https://raw.githubusercontent.com/thynec/CoupCats/refs/heads/main/libraries.R")

# -------------------------- Baseline Data ------------------------------ #

base\_data <- read\_csv("https://www.uky.edu/~clthyn2/base\_data.csv") # Reading in base data.

# 0. Country codes (Thyne 2022).

url <- "https://www.uky.edu/~clthyn2/replace\_ccode\_country.xls" # Bringing in ccodes to merge.

destfile <- "replace\_ccode\_country.xls"

curl::curl\_download(url, destfile)

ccodes <- read\_excel(destfile)

rm(url, destfile)

# -------------------------- Int Signals ----------------------------- #

# 1. WEIS DATA

# 1.1. Getting the data

url <- "https://github.com/thynec/CoupCats/raw/main/weis.fromlai.1966-93.dta" #bringing in the data

weis <- read\_dta(url)

rm(url)

# 1.2. Cleaning

weis <- weis %>%

dplyr::select(year,targcc, mo, gscale) %>% #selecting important variables

rename(

target = targcc,

month = mo

) %>%

mutate("z\_variable" = scale(weis$gscale)) #getting the z score

# 1.3. Aggregating by month # We figured that it is possibly better to do this after we merge both datasets and have them in the same scale

# weis <- weis %>%

#group\_by(year, month, target) %>%

#summarise(across(where(is.numeric), mean, na.rm = TRUE), .groups = "drop") #aggregating by month, year, and target

# 2. COPDAB DATA

# 2.1. Getting the data

library(haven) # we needed this for copdab

url <- "https://www.uky.edu/~clthyn2/copdab.dta"

copdab <- read\_dta(url(url, "rb"))

rm(url)

# 2.2. Cleaning and Fixing year

copdab <- copdab %>%

dplyr::select(year,target, month, value) %>% #selecting important variables

filter(target != 4)

copdab <- copdab %>%

mutate(year = 1900 + year)

# 2.3. Turning values into something meaningful

copdab <- copdab %>%

mutate(weight = case\_when(

value == 15 ~ -102,

value == 14 ~ -65,

value == 13 ~ -50,

value == 12 ~ -44,

value == 11 ~ -29,

value == 10 ~ -16,

value == 9 ~ -6, # Negative for values from 15 to 9

value == 8 ~ 1,

value == 7 ~ 6,

value == 6 ~ 10,

value == 5 ~ 14,

value == 4 ~ 27,

value == 3 ~ 31,

value == 2 ~ 47,

value == 1 ~ 92,

TRUE ~ NA\_real\_ # Ensures NA for any value outside the expected range

)) %>%

select(-value)

copdab <- copdab %>%

mutate("z\_variable" = scale(copdab$weight)) # getting the z score

# -------------------------- V-Dem Data ------------------------------ #

# 1. V-Dem data.

# 1.1. Reading in data.

vdem <- vdem # Code looks strange, but it works.

'The V-Dem dataset does not cover some countries, namely:

Andorra, Antigua and Barbuda, Bahamas, Belize, Brunei, Dominica,

Federated States of Micronesia, Grenada, Kiribati, Liechtenstein,

Marshall Islands, Monaco, Nauru, Palau, Saint Kitts and Nevis,

Saint Lucia, Saint Vincent and the Grenadines, Samoa, San Marino,

Tonga, Tuvalu, and the Vatican.'

# 1.2. Cleaning up data.

vdem <- vdem %>%

subset(select = c(country\_name, # Country.

e\_regionpol\_6C, # Region.

year, # Year.

v2x\_regime, # Regime type.

v2x\_execorr, # Executive corruption index.

v2x\_jucon, # Judicial constraints on the executive index ordinal

v2x\_corr, # Political corruption.

v2smpolsoc, # Polarization of society.

v2cagenmob, # Mass mobilization.

v2x\_civlib, # Civil liberties.

v2x\_rule, # Rule of law.

v2xcs\_ccsi, # Core civil society index.

v2x\_genpp, # Women political participation index.

v2x\_gender, # Women political empowerment index.

v2x\_gencl, # Women civil liberties.

e\_pelifeex, # Life expectancy.

e\_gdp, # GDP.

e\_gdppc, # GDPPC.

e\_miinflat, # Inflation rate.

e\_cow\_exports, # Exports.

e\_cow\_imports, # Imports.

v2smgovshut, # Government Internet shut down in practice.

v2smgovfilprc, # Government Internet filtering in practice,

v2smfordom)) # Foreign governments dissemination of false information.

'Assume these columns are clear of NAs unless stated otherwise.'

vdem <- vdem %>%

rename(country = country\_name,

region = e\_regionpol\_6C,

year = year,

regime = v2x\_regime, # No 1990 for Kazakhstan or Turkmenistan: both gained independence in 1991

exec\_corr = v2x\_execorr, # No data between 1950-2000 for Timor-Leste: gained independence in 2000.

jud\_const = v2x\_jucon, # Ditto Timor-Leste; No 1950-2001 for Bahrain, current government established in 2002.

pol\_corr = v2x\_corr, # Ditto Timor-Leste & Bahrain.

soc\_polar = v2smpolsoc, # Between 2000 to 2023 (no other NAs).

mass\_mobil = v2cagenmob, # No Iceland, Papua New Guinea (2021-2023), Afghanistan (1950-1999), Vietnam (1950-1975), South Yemen (1950-1990).

law\_rule = v2x\_rule, # Ditto Timor-Leste.

civ\_soc = v2xcs\_ccsi,

wom\_polpart = v2x\_genpp, # Lots of issues... run: na\_df <- vdem[is.na(vdem$wom\_polpart), , drop = FALSE]

women\_polemp = v2x\_gender, # Run: na\_df <- vdem[is.na(vdem$v2x\_gender), , drop = FALSE]

wom\_civlib = v2x\_gencl, # No issues--could replace wom\_polpart and women\_polemp

life\_exp = e\_pelifeex, # Between 1800 to 2022; missing South Yemen, Republic of Vietnam (1950-75), Kosovo, German Democratic Republic, Palestine/Gaza, Somaliland, Hong Kong, Zanzibar

gdp = e\_gdp, # Between 1789 to 2019... good for past data, might need something else for current data.

gdppc = e\_gdppc, # Ditto GDP.

infla\_rate = e\_miinflat, # Ditto GDP.

exports = e\_cow\_exports, # Between 1870 to 2014.

imports = e\_cow\_imports, # Between 1870 to 2014.

int\_shutdown = v2smgovshut, # Between 2000 to 2023... can likely code this to be 0 for previous years.

int\_censor = v2smgovfilprc, # Between 2000 to 2023.

forgov\_misinfo = v2smfordom) %>% # Between 2000 to 2023.

filter(year >= 1950)

vdem <- vdem %>% # Merging in ccodes.

left\_join(ccodes, by = c("year", "country")) %>%

select(ccode, everything()) %>%

subset(select = -c(country))

# 1.3. Merging into data set.

base\_data <- base\_data %>%

left\_join(vdem, by = c("ccode", "year")) # No duplicates.

# -------------------------- Political Data ------------------------------ #

# 1. Perceptions of Corruption (Transparency International, Corruption Perceptions Index).

# 1.1. Reading in data.

wdi\_data <- WDI(indicator = c("CC.EST", "CC.PER.RNK"), country = "all", start = 1996, end = 2023, extra = TRUE) # Data are reported every two years--need to figure out a way to transform.

# 1.2. Cleaning up data.

wdi\_data <- wdi\_data %>%

mutate(year=year+1) %>% #just lagged by adding a year

subset(select = c(country, year, CC.EST, CC.PER.RNK, region)) %>%

rename(corruption\_est = CC.EST,

corruption\_rank = CC.PER.RNK,

wdi\_region = region) %>%

set\_variable\_labels(

corruption\_est="Control of Corruption: Estimate; from WDI; t-1",

corruption\_rank="Control of Corruption: Percentile Rank; from WDI; t-1",

wdi\_region="regions from WDI"

)

wdi\_data <- wdi\_data %>%

left\_join(ccodes, by = c("year", "country"))

check <- wdi\_data %>%

filter(if\_any(everything(), is.na))

table(check$country)

table(check$year)

#check looks fine; missing 3 in normal places due to temporal coverage; other missings are due to small countries or regions

rm(check)

# 1.3. Merging into data set. Do this twice. First for corruption by ccode/year; then for region by ccode (region doesn't vary by time)

wdi1 <- wdi\_data %>%

select(-country, -wdi\_region)

base\_data <- base\_data %>%

left\_join(wdi1, by = c("year", "ccode")) # NAs are unreported years.

wdi2 <- wdi\_data %>%

select(ccode, wdi\_region) %>%

distinct()

base\_data <- base\_data %>%

left\_join(wdi2, by=c("ccode"))

rm(wdi\_data, wdi1, wdi2)

#Add regional dummies

table(base\_data$wdi\_region)

base\_data <- base\_data %>%

mutate(e\_asia\_pacific=ifelse(wdi\_region=="East Asia & Pacific", 1, 0)) %>%

mutate(euro\_cent\_asia=ifelse(wdi\_region=="Europe & Central Asia", 1, 0)) %>%

mutate(LA\_carrib=ifelse(wdi\_region=="Latin America & Caribbean", 1, 0)) %>%

mutate(MENA=ifelse(wdi\_region=="Middle East & North Africa", 1, 0)) %>%

mutate(N\_america=ifelse(wdi\_region=="North America", 1, 0)) %>%

mutate(S\_asia=ifelse(wdi\_region=="South Asia", 1, 0)) %>%

mutate(Sub\_africa=ifelse(wdi\_region=="Sub-Saharan Africa", 1, 0)) %>%

set\_variable\_labels(

e\_asia\_pacific="East Asia & Pacific",

euro\_cent\_asia="Europe & Central Asia",

LA\_carrib="Latin America & Caribbean",

MENA="Middle East & North Africa",

N\_america="North America",

S\_asia="South Asia",

Sub\_africa="Sub-Saharan Africa") %>%

select(-wdi\_region)

# 2. Regime data (V-Dem)

# 2.1. Reading in data.

url <- "https://github.com/vdeminstitute/vdemdata/raw/master/data/vdem.RData"

destfile <- tempfile(fileext = ".RData")

download.file(url, destfile, mode = "wb")

load(destfile)

rm(destfile, url)

# 2.2. Cleaning up data.

regime\_type <- vdem %>%

subset(select = c(country\_name, year, v2x\_regime)) %>%

rename(country = country\_name,

regime\_type = v2x\_regime) %>%

filter(year >= 1950)

regime\_type <- regime\_type %>%

left\_join(ccodes, by = c("country", "year")) %>% # NAs resulting from state-like actors, not full states.

subset(select = -c(country)) %>% # To prevent future duplicated columns.

drop\_na() # No duplicates either!

rm(vdem)

label(regime\_type$regime\_type) <- "0 = Closed autocracy, 1 = Electoral autocracy, 2 = Electoral democracy, 3 = Liberal Democracy"

# 2.3 Organizing variables for regression

regime\_type <- regime\_type %>%

mutate(

closed\_autocracy = ifelse(regime\_type == 0, 1, 0),

electoral\_autocracy = ifelse(regime\_type == 1, 1, 0),

electoral\_democracy = ifelse(regime\_type == 2, 1, 0),

liberal\_democracy = ifelse(regime\_type == 3, 1, 0)

)

# 2.4. Merging into data set.

base\_data <- base\_data %>%

left\_join(regime\_type, by = c("ccode", "year")) # Missing data simply is not updated by V-Dem, so I will not be dropping them.

rm(regime\_type)

# -------------------------- Social Data ------------------------------ #

# 1. Coup data (Powell & Thyne 2011).

# 1.1. Reading in data.

coup\_data <- read\_delim("http://www.uky.edu/~clthyn2/coup\_data/powell\_thyne\_coups\_final.txt",

delim = "\t", escape\_double = FALSE,

trim\_ws = TRUE)

# 1.2. Cleaning up data.

check <- coup\_data %>%

filter(ccode==lag(ccode) & year==lag(year) & month==lag(month))

#we have 6 cases of 2 coups in the same month:

#Haiti, 41, 04/1989; both failed

#Bolivia, 145, 05/1981; both failed

#Argentina, 160, 12/1975; both failed

#Sierra Leone, 451, 03/1967; second successful

#Togo, 461, 10/1991; both failed

#Sudan, 625, 12/1966; both failed

#Just need to make successful in Sierra Leone trump the failed one; all others are fine because both failed

table(coup\_data$coup)

coup\_data <- coup\_data %>%

mutate(coup=ifelse(ccode==451 & year==1967 & month==3, 2, coup))

table(coup\_data$coup) #looks good; proceed...

rm(check)

coup\_data <- coup\_data %>%

select(-ccode\_gw, -ccode\_polity, -day, -version) %>%

distinct() %>% #dropped the obs where 2 coups in same month, as above

mutate(coup\_attempt = 1) %>%

mutate(coup\_successful=ifelse(coup==2, 1, NA)) %>%

mutate(coup\_failed=ifelse(coup==1, 1, NA))

table(coup\_data$coup\_attempt)

table(coup\_data$coup, coup\_data$coup\_attempt)

table(coup\_data$coup, coup\_data$coup\_successful)

table(coup\_data$coup, coup\_data$coup\_failed)

#after merge, should have...

#487 attempts

#246 successful

#241 failed

#everything above looks good; proceed...

# 1.3. Merging into data set.

coup\_data <- coup\_data %>%

select(-country, -coup)

base\_data <- base\_data %>%

left\_join(coup\_data, by = c("year", "ccode", "month"))

base\_data <- base\_data %>%

mutate(coup\_attempt = ifelse(is.na(coup\_attempt) & year>=1950, 0, as.numeric(coup\_attempt))) %>%

mutate(coup\_successful = ifelse(is.na(coup\_successful) & year>=1950, 0, as.numeric(coup\_successful))) %>%

mutate(coup\_failed = ifelse(is.na(coup\_failed) & year>=1950, 0, as.numeric(coup\_failed)))

table(base\_data$coup\_attempt)

table(base\_data$coup\_successful)

table(base\_data$coup\_failed)

#we lost one successful coup because COW doesn't recognize Oman in 1970; we're good on this; proceed...

rm(coup\_data) # Keeping things clean

# 1.4. Set up non-coup months to deal with autocorrelation in the analyses

base\_data <- base\_data %>%

arrange(ccode, year, month) %>%

group\_by(ccode) %>%

mutate(sequence=row\_number())

base\_data <- base\_data %>%

btscs('coup\_attempt', 'year', 'ccode') %>%

rename(pce=spell\_time) %>%

mutate(pce2=pce\*pce) %>%

mutate(pce3=pce\*pce\*pce) %>%

set\_variable\_labels(

pce="Months since last coup",

pce2="Months^2",

pce3="Months^3") %>%

select(-sequence)

#For above, peace years are set up ignoring success/failed; go back and re-created these if you end up wanting to analyze success/failed instead of all attempts

# 2. Population data (World Bank Data Group 2024).

# 2.1. Reading in data.

url <- "https://extdataportal.worldbank.org/content/dam/sites/data/gender-data/data/data-gen/zip/indicator/population-number.zip"

zip\_file <- "population-data.zip"

download.file(url, zip\_file, mode = "wb")

unzip(zip\_file, exdir = "unzipped\_data")

world\_bank <- read.csv("unzipped\_data/Population (number).csv") # WB is missing data for 2024. This will update automatically after they add it.

rm(url, zip\_file)

unlink("population-data.zip")

unlink("unzipped\_data", recursive=TRUE)

# 2.2. Cleaning up data.

pop <- world\_bank %>%

filter(Indicator.Name == 'Population, total') %>%

subset(select = c(Country.Name, Year, Value)) %>%

rename(country = Country.Name,

year = Year,

population = Value) %>%

left\_join(ccodes, by = c("year", "country"))

check <- pop %>%

filter(if\_any(everything(), is.na))

table(check$country) #all regions or tiny countries; we're fine to drop NAs

pop <- pop %>%

drop\_na() %>%

select(-country)

rm(check)

check <- pop %>%

distinct() #no duplicates in pop data

check <- base\_data %>%

distinct() #no duplicates in base data

rm(check)

pop <- pop %>%

mutate(year=year+1) %>%

mutate(pop=log(population)) %>%

set\_variable\_labels(

pop="Total pop, WDI, log, t-1"

) %>%

select(-population)

# 2.3. Merging into data set.

base\_data <- base\_data %>%

ungroup() %>%

left\_join(pop, by = c("ccode", "year"))

rm(pop)

check <- base\_data %>%

filter(is.na(pop))

histogram(check$year) #looks fine but should be able to get pop for full sample from elsewhere

rm(check)

rm(world\_bank)

# 4. Median age data (World Bank Data Group 2024)

# 4.1. Reading in data.

url <- "https://ourworldindata.org/grapher/median-age.csv?v=1&csvType=full&useColumnShortNames=true"

median\_age <- read\_csv(url)

rm(url)

# 4.2. Cleaning up data.

median\_age <- median\_age %>%

subset(select = c(Entity, Year, median\_age\_\_sex\_all\_\_age\_all\_\_variant\_estimates, median\_age\_\_sex\_all\_\_age\_all\_\_variant\_medium)) %>%

rename(country = Entity,

year = Year,

popln\_median = median\_age\_\_sex\_all\_\_age\_all\_\_variant\_estimates,

popln\_median\_est = median\_age\_\_sex\_all\_\_age\_all\_\_variant\_medium) %>%

mutate(median\_age = ifelse(is.na(popln\_median), popln\_median\_est, popln\_median)) %>%

subset(select = -c(popln\_median, popln\_median\_est)) %>%

filter(year <= 2024)

median\_age <- median\_age %>%

mutate(year=year+1) %>% #just lagged

left\_join(ccodes, by = c("country", "year"))

check <- median\_age %>%

filter(is.na(ccode))

table(check$country) #fine; all regions or very small countries

rm(check)

median\_age <- median\_age %>%

drop\_na() %>% # NAs from non-country rows.

distinct() %>% # Random duplicates.

select(-country)

# 4.3. Merging into data set.

base\_data <- base\_data %>%

left\_join(median\_age, by = c("ccode", "year"))

rm(median\_age)

# ------------------------------------ Military data ------------------------------------ #

# 1.Military expenditure (% of GDP)

# 1.1 Loading data

url <- "https://api.worldbank.org/v2/en/indicator/MS.MIL.XPND.GD.ZS?downloadformat=excel"

destfile <- "MS\_MIL\_XPND\_GD.xls"

curl::curl\_download(url, destfile)

milex <- read\_excel(destfile, skip = 3)

rm(destfile, url)

#1.2 Reshaping data

milex <- milex %>%

rename( "country" = `Country Name`) %>%

subset(select = -c(`Indicator Name`, `Indicator Code`, `Country Code`)) #removing things we don't want

milex <- milex %>%

pivot\_longer(

cols = -c(country), # Keep country-related columns fixed

names\_to = "year",

values\_to = "military expenditure"

) %>%

mutate(year = as.integer(year)) # Convert Year to integer

#merge to base

milex <- milex %>%

rename(milex='military expenditure') %>%

set\_variable\_labels(milex="Milit expenditure, % of GDP") %>%

mutate(year=year+1) #lagged

milex <- milex %>%

left\_join(ccodes, by=c("country", "year")) %>%

select(-country) %>%

distinct()

base\_data <- base\_data %>%

left\_join(milex, by=c("ccode", "year"))

rm(milex)

# 2.Armed forces personnel (total)

# 2.1 Loading data

url <- "https://api.worldbank.org/v2/en/indicator/MS.MIL.TOTL.P1?downloadformat=excel"

destfile <- "MS\_MIL\_TOTL.xls"

curl::curl\_download(url, destfile)

milper <- read\_excel(destfile, skip = 3)

rm(destfile, url)

# 2.2 Reshaping data; cleaning a bit

milper <- milper %>%

rename( "country" = `Country Name`) %>%

subset(select = -c(`Indicator Name`, `Indicator Code`, `Country Code`)) #removing things we don't want

milper <- milper %>%

pivot\_longer(

cols = -c(country), # Keep country-related columns fixed

names\_to = "year",

values\_to = "milper"

) %>%

mutate(year = as.integer(year)) # Convert Year to integer

milper <- milper %>%

drop\_na() %>%

mutate(milper=log(milper+1)) %>% #just logged

mutate(year=year+1) %>% #just lagged

set\_variable\_labels(milper="milit personell; logged; t-1")

# 3 Merging

milper <- milper %>%

left\_join(ccodes, by=c("country", "year")) %>%

drop\_na() %>%

select(-country)

base\_data <- base\_data %>%

left\_join(milper, by=c("ccode", "year"))

rm(milper)

###############################################################################################

#Checked through above and ready to produce .csv and upload to github

write.csv(base\_data, gzfile("base\_data.csv.gz"), row.names = FALSE)

#Now push push the file that was just written to the working directory to github

###############################################################################################

#### Filling in NAs below, committed on 3/7 ####

# Source the interpolation functions script

source("interpolation\_functions.R")

# Load the dataset

url <- "https://raw.githubusercontent.com/thynec/CoupCats/refs/heads/main/base\_data.csv"

df <- read.csv(url)

# Check for missing values

values <- check\_missing\_values(df)

# Fill missing values using forward/backward fill

df <- fill\_missing\_values(df, values)

# Interpolate missing values

for (val in values) {

df <- interpolate\_missing\_values(df, val, max\_gap = 36, method = 'spline', order = 3)

}

# Check remaining missing values

cat("\nFinal missing values check:\n")

for (col in c(values, paste0(values, '\_interpolated'))) {

if (col %in% colnames(df)) {

na\_count <- sum(is.na(df[[col]]))

if (na\_count > 0) {

total <- nrow(df)

pct <- na\_count / total \* 100

cat(sprintf("%s has %d / %d rows (%.2f%%) missing values\n", col, na\_count, total, pct))

}

}

}

#### End of Commit ####

###############################################################################################

#Below is stuff we probably won't use but keeping it here just in case...

###############################################################################################

# 3. Age population data (World Bank Data Group 2024).

# 3.1. Reading in data.

#age\_popln <- world\_bank

#%>%

# 3.2. Cleaning up data.

# filter(Indicator.Name %in% c('Population ages 0-14, total',

# 'Population ages 15-64, total',

# 'Population ages 65 and above, total')) %>%

# subset(select = c(Country.Name, Year, Value, Disaggregation)) %>%

# rename(country = Country.Name,

# year = Year,

# age\_tot = Value,

# type = Disaggregation)

#age\_popln <- age\_popln %>%

# left\_join(ccodes, by = c("country", "year"), relationship = "many-to-many") %>%

# drop\_na() %>% # NAs from non-country rows.

# distinct() # Random duplicates.

#age\_popln <- age\_popln %>%

# pivot\_wider(names\_from = type, values\_from = age\_tot, values\_fn = first) %>%

# rename(age0\_14 = `total, 0-14`,

# age15\_64 = `total, 15-64`,

# age65plus = `total, 65+`)

# 3.3. Merging into data set.

#emma\_data <- emma\_data %>% # No 2024 data.

# left\_join(age\_popln, by = c("country", "year", "ccode"), relationship = "many-to-many") %>%

# distinct() # Random duplicates.

#rm(world\_bank, age\_popln)

# 5. Age expectancy data.

# 5.1. Reading in data.

#url <- "https://srhdpeuwpubsa.blob.core.windows.net/whdh/DATADOT/INDICATOR/C64284D\_ALL\_LATEST.csv"

#age\_expectancy <- read\_csv(url)

#rm(url)

# 5.2. Cleaning up data.

#age\_expectancy <- age\_expectancy %>%

# subset(select = c(DIM\_TIME, GEO\_NAME\_SHORT, DIM\_SEX, AMOUNT\_N)) %>%

# filter(DIM\_SEX == 'TOTAL') %>%

# rename(country = GEO\_NAME\_SHORT,

# year = DIM\_TIME,

# age\_expectancy = AMOUNT\_N) %>%

# subset(select = -c(DIM\_SEX))

#age\_expectancy <- age\_expectancy %>%

# left\_join(ccodes, by = c("country", "year")) %>% # Missing 'Côte d'Ivoire', 'Türkiye', 'Zambia', 'Zimbabwe'

# drop\_na() %>% # NAs from non-country rows.

# distinct() # Random duplicates from Yemen.

# 5.3. Merging into data set.

#emma\_data <- emma\_data %>%

# left\_join(age\_expectancy, by = c("country", "year", "ccode"), relationship = "many-to-many")

#rm(age\_expectancy)

rm(list = ls())

setwd("~/R/coupcats") # Set working file.

# source("https://raw.githubusercontent.com/thynec/CoupCats/refs/heads/main/packages.R")

source("https://raw.githubusercontent.com/thynec/CoupCats/refs/heads/main/libraries.R")

source("https://raw.githubusercontent.com/thynec/CoupCats/refs/heads/main/step1.R")

# -------------------------- V-Dem Data ------------------------------ #

# V-Dem data.

# Reading in data.

vdem <- vdem

'The V-Dem dataset does not cover some countries, namely:

Andorra, Antigua and Barbuda, Bahamas, Belize, Brunei, Dominica,

Federated States of Micronesia, Grenada, Kiribati, Liechtenstein,

Marshall Islands, Monaco, Nauru, Palau, Saint Kitts and Nevis,

Saint Lucia, Saint Vincent and the Grenadines, Samoa, San Marino,

Tonga, Tuvalu, and the Vatican.'

# Cleaning up data.

vdem <- vdem %>%

subset(select = c(country\_name, # Country.

e\_regionpol\_6C, # Region.

year, # Year.

v2x\_regime, # Regime type.

v2x\_polyarchy, # Regime type.

v2x\_execorr, # Executive corruption index.

v2x\_jucon, # Judicial constraints on the executive index ordinal

v2x\_corr, # Political corruption.

v2cacamps, # Political polarization.

v2x\_civlib, # Civil liberties.

v2x\_rule, # Rule of law.

v2xcs\_ccsi, # Core civil society index.

v2x\_genpp, # Women political participation index.

v2x\_gender, # Women political empowerment index.

v2x\_gencl, # Women civil liberties.

e\_pelifeex, # Life expectancy.

e\_gdp, # GDP.

e\_gdppc, # GDPPC.

e\_miinflat, # Inflation rate.

e\_cow\_exports, # Exports.

e\_cow\_imports, # Imports.

v2smgovshut, # Government Internet shut down in practice.

v2smgovfilprc, # Government Internet filtering in practice,

v2smfordom)) # Foreign governments dissemination of false information.

'Assume these columns are clear of NAs unless stated otherwise.'

vdem <- vdem %>%

rename(country = country\_name,

region = e\_regionpol\_6C,

year = year,

regime = v2x\_regime,

regime2 = v2x\_polyarchy,

exec\_corr = v2x\_execorr,

jud\_const = v2x\_jucon,

pol\_corr = v2x\_corr,

civ\_lib = v2x\_civlib,

pol\_polar = v2cacamps,

law\_rule = v2x\_rule, # Ditto Timor-Leste.

civ\_soc = v2xcs\_ccsi,

wom\_polpart = v2x\_genpp, # Lots of issues... run: na\_df <- vdem[is.na(vdem$wom\_polpart), , drop = FALSE]

women\_polemp = v2x\_gender, # Run: na\_df <- vdem[is.na(vdem$v2x\_gender), , drop = FALSE]

wom\_civlib = v2x\_gencl, # No issues--could replace wom\_polpart and women\_polemp

life\_exp = e\_pelifeex, # Between 1800 to 2022; missing South Yemen, Republic of Vietnam (1950-75), Kosovo, German Democratic Republic, Palestine/Gaza, Somaliland, Hong Kong, Zanzibar

gdp = e\_gdp, # Between 1789 to 2019... good for past data, might need something else for current data.

gdppc = e\_gdppc, # Ditto GDP.

infla\_rate = e\_miinflat, # Ditto GDP.

exports = e\_cow\_exports, # Between 1870 to 2014.

imports = e\_cow\_imports, # Between 1870 to 2014.

int\_shutdown = v2smgovshut, # Between 2000 to 2023... can likely code this to be 0 for previous years.

int\_censor = v2smgovfilprc, # Between 2000 to 2023.

forgov\_misinfo = v2smfordom) %>% # Between 2000 to 2023.

filter(year >= 1950)

vdem <- vdem %>% # Merging in ccodes.

left\_join(ccodes, by = c("year", "country")) %>%

filter(!is.na(ccode)) # Non-state actors.

# GDPPC (e\_gdppc).

# Cleaning data.

vdem\_gdppc <- vdem %>%

subset(select = c(country, ccode, year, gdppc)) %>%

rename(vdem\_gdppc = gdppc,

c\_merge = country)

base\_data <- base\_data %>%

left\_join(vdem\_gdppc, by = c("ccode", "year"))

check <- base\_data %>%

subset(select = c(country, c\_merge, year)) %>%

distinct() %>%

filter(country!=c\_merge)

rm(check) # All good.

base\_data <- base\_data %>%

subset(select = -c(c\_merge))

rm(vdem\_gdppc)

# Reading in WDI.

wdi\_gdppc <- WDI(country = "all",

indicator = "NY.GDP.PCAP.CD",

start = 1960,

end = 2024,

extra = TRUE)

wdi\_gdppc <- wdi\_gdppc %>%

subset(select = c(country, year, NY.GDP.PCAP.CD)) %>%

rename(wdi\_gdppc = NY.GDP.PCAP.CD) %>%

left\_join(ccodes, by = c("year", "country")) %>%

rename(c\_merge = country)

base\_data <- base\_data %>%

left\_join(wdi\_gdppc, by = c("ccode", "year"))

check <- base\_data %>%

subset(select = c(country, c\_merge)) %>%

distinct() %>%

filter(country!=c\_merge)

rm(check) # All good.

base\_data <- base\_data %>%

subset(select = -c(c\_merge))

rm(wdi\_gdppc)

# 1. Regime type (v2x\_polyarchy).

vdem\_regime2 <- vdem %>%

subset(select = c(country, ccode, year, regime2)) %>%

filter(!(country == "Kazakhstan" & year == 1990)) %>% # Kazakhstan became independent this year.

filter(!(country == "Turkmenistan" & year == 1990)) %>% # Turkmenistan became independent this year.

subset(select = -c(country))

base\_data <- base\_data %>%

left\_join(vdem\_regime2, by = c("ccode", "year")) %>%

filter(!(ccode %in% c(232, 58, 31, 80, 835, 54, 987, 55, 946, 223,

983, 221, 970, 986, 60, 56, 57, 990, 331, 955,

947))) %>%

filter(!(ccode == "471" & year == 1960)) %>% # Cameroon became independent this year.

filter(!(ccode == "678" & year <= 1991)) %>% # Yemen unified in 1991.

filter(year < 2024)

base\_data <- base\_data %>% # Ultimately, we are losing Czechoslovakia and the German Federal Republic.

filter(!(ccode == "260")) %>%

filter(!(ccode == "315"))

rm(vdem\_regime2)

# Interpolating by WDI.

base\_data <- base\_data %>%

select(country, ccode, year, vdem\_gdppc, wdi\_gdppc) %>%

distinct() %>%

mutate(change=(wdi\_gdppc-lag(wdi\_gdppc))) %>%

mutate(now=vdem\_gdppc) %>%

mutate(perc\_change=change/lag(wdi\_gdppc)) %>%

mutate(gdppc=ifelse(is.na(now), lag(now)\*perc\_change+lag(vdem\_gdppc), now)) %>%

mutate(gdppc=ifelse(is.na(gdppc), lag(gdppc)\*perc\_change+lag(gdppc), gdppc)) %>%

mutate(gdppc=ifelse(is.na(gdppc), lag(gdppc)\*perc\_change+lag(gdppc), gdppc)) %>%

subset(select = -c(wdi\_gdppc, vdem\_gdppc, change, now, perc\_change))

# 2. Regime type (v2x\_regime).

vdem\_regime <- vdem %>%

subset(select = c(country, ccode, year, regime)) %>%

filter(!(country == "Kazakhstan" & year == 1990)) %>%

filter(!(country == "Turkmenistan" & year == 1990)) %>%

subset(select = -c(country))

base\_data <- base\_data %>%

left\_join(vdem\_regime, by = c("ccode", "year")) # No duplicates.

rm(vdem\_regime)

# 3. Executive corruption index (v2x\_execorr).

vdem\_execorr <- vdem %>%

subset(select = c(country, ccode, year, exec\_corr)) %>%

filter(!(ccode == "860" & year < 2001)) %>% # Timor-Leste gained independence in 2000.

subset(select = -c(country))

base\_data <- base\_data %>%

left\_join(vdem\_execorr, by = c("ccode", "year"))

rm(vdem\_execorr)

# 4. Judicial constraints on the executive index (v2x\_jucon)

vdem\_judconst <- vdem %>%

subset(select = c(country, ccode, year, jud\_const)) %>%

filter(!(ccode == "860" & year < 2001)) %>%

subset(select = -c(country))

base\_data <- base\_data %>%

left\_join(vdem\_judconst, by = c("ccode", "year")) %>%

filter(!(ccode == "692" & year < 2002)) # Current government of Bahrain established in 2002.

rm(vdem\_judconst)

# 5. Political corruption (v2x\_corr).

vdem\_polcorr <- vdem %>%

subset(select = c(country, ccode, year, pol\_corr)) %>%

filter(!(ccode == "860" & year < 2001)) %>%

filter(!(ccode == "692" & year < 2002))

# We need to decide what to do with Bahrain 2002-2004...

rm(vdem\_polcorr)

# 6. Polarization of society (v2smpolsoc).

vdem\_polpolar <- vdem %>%

subset(select = c(country, ccode, year, pol\_polar))

# No South Yemen (1950-90), Republic of Vietnam (1950-75), Afghanistan (1950-1991), Papua New Guinea (2021-23), Comoros (2023), Iceland (1950-1991)

rm(vdem\_polpolar)

# 7. Civil liberties (v2x\_civlib).

vdem\_civlib <- vdem %>%

subset(select = c(country, ccode, year, civ\_lib)) %>%

subset(select = -c(country))

base\_data <- base\_data %>%

left\_join(vdem\_civlib, by = c("ccode", "year"))

rm(vdem\_civlib)

# 8. Women political participation index (v2x\_genpp).

df <- vdem %>%

subset(select = c(country, ccode, year, wom\_polpart))

# Women political empowerment index (v2x\_gender).

# Women civil liberties (v2x\_gencl).

df\_na <- df[rowSums(is.na(df)) > 0, ]

# df\_na <- df[rowSums(is.na(df)) > 0, ]

# df\_dup <- df[duplicated(df) | duplicated(df, fromLast = TRUE), ]

###########################################################################

####READ ME!

#CoupCats - please skim to look for "####CT:..." each time you open this. I'm making notes as I clean things to help you learn some stuff. Once you read the note go ahead and delete it.

###########################################################################

rm(list=ls())

#setwd("C:/Users/Camila/OneDrive/R related/R/R") # Set working file. #Camila

setwd("C:/Users/clayt/OneDrive - University of Kentucky/elements/current\_research/coupcats") #Clay at home

#load packages: if you use new packages, try to remember to add them to the packages on github. Otherwise, just flag them and I'll do it.

#source("https://raw.githubusercontent.com/thynec/CoupCats/refs/heads/main/packages.R")

#load libraries: if you add new libraries, try to remember to add them to github. Otherwise, just flag them and I'll do it.

source("https://raw.githubusercontent.com/thynec/CoupCats/refs/heads/main/libraries.R")

# -------------------------- Baseline Data ------------------------------ #

base\_data <- read\_csv("https://www.uky.edu/~clthyn2/base\_data.csv") # Reading in base data.

# 0. Country codes (Thyne 2022).

url <- "https://www.uky.edu/~clthyn2/replace\_ccode\_country.xls" # Bringing in ccodes to merge.

destfile <- "replace\_ccode\_country.xls"

curl::curl\_download(url, destfile)

ccodes <- read\_excel(destfile)

rm(url, destfile)

# -------------------------- Int Signals ----------------------------- #

# 1. WEIS DATA

# 1.1. Getting the data

url <- "https://github.com/thynec/CoupCats/raw/main/weis.fromlai.1966-93.dta" #bringing in the data

weis <- read\_dta(url)

rm(url)

# 1.2. Cleaning

weis <- weis %>%

dplyr::select(year,targcc, mo, gscale) %>% #selecting important variables

rename(

target = targcc,

month = mo

) %>%

mutate("z\_variable" = scale(weis$gscale)) #getting the z score

# 1.3. Aggregating by month # We figured that it is possibly better to do this after we merge both datasets and have them in the same scale

# weis <- weis %>%

#group\_by(year, month, target) %>%

#summarise(across(where(is.numeric), mean, na.rm = TRUE), .groups = "drop") #aggregating by month, year, and target

# 2. COPDAB DATA

# 2.1. Getting the data

library(haven) # we needed this for copdab

url <- "https://www.uky.edu/~clthyn2/copdab.dta"

copdab <- read\_dta(url(url, "rb"))

rm(url)

# 2.2. Cleaning and Fixing year

copdab <- copdab %>%

dplyr::select(year,target, month, value) %>% #selecting important variables

filter(target != 4)

copdab <- copdab %>%

mutate(year = 1900 + year)

# 2.3. Turning values into something meaningful

copdab <- copdab %>%

mutate(weight = case\_when(

value == 15 ~ -102,

value == 14 ~ -65,

value == 13 ~ -50,

value == 12 ~ -44,

value == 11 ~ -29,

value == 10 ~ -16,

value == 9 ~ -6, # Negative for values from 15 to 9

value == 8 ~ 1,

value == 7 ~ 6,

value == 6 ~ 10,

value == 5 ~ 14,

value == 4 ~ 27,

value == 3 ~ 31,

value == 2 ~ 47,

value == 1 ~ 92,

TRUE ~ NA\_real\_ # Ensures NA for any value outside the expected range

)) %>%

select(-value)

copdab <- copdab %>%

mutate("z\_variable" = scale(copdab$weight)) # getting the z score

# -------------------------- V-Dem Data ------------------------------ #

# 1. V-Dem data.

# 1.1. Reading in data.

vdem <- vdem # Code looks strange, but it works.

'The V-Dem dataset does not cover some countries, namely:

Andorra, Antigua and Barbuda, Bahamas, Belize, Brunei, Dominica,

Federated States of Micronesia, Grenada, Kiribati, Liechtenstein,

Marshall Islands, Monaco, Nauru, Palau, Saint Kitts and Nevis,

Saint Lucia, Saint Vincent and the Grenadines, Samoa, San Marino,

Tonga, Tuvalu, and the Vatican.'

# 1.2. Cleaning up data.

vdem <- vdem %>%

subset(select = c(country\_name, # Country.

e\_regionpol\_6C, # Region.

year, # Year.

v2x\_regime, # Regime type.

v2x\_execorr, # Executive corruption index.

v2x\_jucon, # Judicial constraints on the executive index ordinal

v2x\_corr, # Political corruption.

v2smpolsoc, # Polarization of society.

v2cagenmob, # Mass mobilization.

v2x\_civlib, # Civil liberties.

v2x\_rule, # Rule of law.

v2xcs\_ccsi, # Core civil society index.

v2x\_genpp, # Women political participation index.

v2x\_gender, # Women political empowerment index.

v2x\_gencl, # Women civil liberties.

e\_pelifeex, # Life expectancy.

e\_gdp, # GDP.

e\_gdppc, # GDPPC.

e\_miinflat, # Inflation rate.

e\_cow\_exports, # Exports.

e\_cow\_imports, # Imports.

v2smgovshut, # Government Internet shut down in practice.

v2smgovfilprc, # Government Internet filtering in practice,

v2smfordom)) # Foreign governments dissemination of false information.

'Assume these columns are clear of NAs unless stated otherwise.'

vdem <- vdem %>%

rename(country = country\_name,

region = e\_regionpol\_6C,

year = year,

regime = v2x\_regime, # No 1990 for Kazakhstan or Turkmenistan: both gained independence in 1991

exec\_corr = v2x\_execorr, # No data between 1950-2000 for Timor-Leste: gained independence in 2000.

jud\_const = v2x\_jucon, # Ditto Timor-Leste; No 1950-2001 for Bahrain, current government established in 2002.

pol\_corr = v2x\_corr, # Ditto Timor-Leste & Bahrain.

soc\_polar = v2smpolsoc, # Between 2000 to 2023 (no other NAs).

mass\_mobil = v2cagenmob, # No Iceland, Papua New Guinea (2021-2023), Afghanistan (1950-1999), Vietnam (1950-1975), South Yemen (1950-1990).

law\_rule = v2x\_rule, # Ditto Timor-Leste.

civ\_soc = v2xcs\_ccsi,

wom\_polpart = v2x\_genpp, # Lots of issues... run: na\_df <- vdem[is.na(vdem$wom\_polpart), , drop = FALSE]

women\_polemp = v2x\_gender, # Run: na\_df <- vdem[is.na(vdem$v2x\_gender), , drop = FALSE]

wom\_civlib = v2x\_gencl, # No issues--could replace wom\_polpart and women\_polemp

life\_exp = e\_pelifeex, # Between 1800 to 2022; missing South Yemen, Republic of Vietnam (1950-75), Kosovo, German Democratic Republic, Palestine/Gaza, Somaliland, Hong Kong, Zanzibar

gdp = e\_gdp, # Between 1789 to 2019... good for past data, might need something else for current data.

gdppc = e\_gdppc, # Ditto GDP.

infla\_rate = e\_miinflat, # Ditto GDP.

exports = e\_cow\_exports, # Between 1870 to 2014.

imports = e\_cow\_imports, # Between 1870 to 2014.

int\_shutdown = v2smgovshut, # Between 2000 to 2023... can likely code this to be 0 for previous years.

int\_censor = v2smgovfilprc, # Between 2000 to 2023.

forgov\_misinfo = v2smfordom) %>% # Between 2000 to 2023.

filter(year >= 1950)

vdem <- vdem %>% # Merging in ccodes.

left\_join(ccodes, by = c("year", "country")) %>%

select(ccode, everything()) %>%

subset(select = -c(country))

# 1.3. Merging into data set.

base\_data <- base\_data %>%

left\_join(vdem, by = c("ccode", "year")) # No duplicates.

# -------------------------- Political Data ------------------------------ #

# 1. Perceptions of Corruption (Transparency International, Corruption Perceptions Index).

# 1.1. Reading in data.

wdi\_data <- WDI(indicator = c("CC.EST", "CC.PER.RNK"), country = "all", start = 1996, end = 2023, extra = TRUE) # Data are reported every two years--need to figure out a way to transform.

# 1.2. Cleaning up data.

wdi\_data <- wdi\_data %>%

mutate(year=year+1) %>% #just lagged by adding a year

subset(select = c(country, year, CC.EST, CC.PER.RNK, region)) %>%

rename(corruption\_est = CC.EST,

corruption\_rank = CC.PER.RNK,

wdi\_region = region) %>%

set\_variable\_labels(

corruption\_est="Control of Corruption: Estimate; from WDI; t-1",

corruption\_rank="Control of Corruption: Percentile Rank; from WDI; t-1",

wdi\_region="regions from WDI"

)

wdi\_data <- wdi\_data %>%

left\_join(ccodes, by = c("year", "country"))

check <- wdi\_data %>%

filter(if\_any(everything(), is.na))

table(check$country)

table(check$year)

#check looks fine; missing 3 in normal places due to temporal coverage; other missings are due to small countries or regions

rm(check)

# 1.3. Merging into data set. Do this twice. First for corruption by ccode/year; then for region by ccode (region doesn't vary by time)

wdi1 <- wdi\_data %>%

select(-country, -wdi\_region)

base\_data <- base\_data %>%

left\_join(wdi1, by = c("year", "ccode")) # NAs are unreported years.

wdi2 <- wdi\_data %>%

select(ccode, wdi\_region) %>%

distinct()

base\_data <- base\_data %>%

left\_join(wdi2, by=c("ccode"))

rm(wdi\_data, wdi1, wdi2)

#Add regional dummies

table(base\_data$wdi\_region)

base\_data <- base\_data %>%

mutate(e\_asia\_pacific=ifelse(wdi\_region=="East Asia & Pacific", 1, 0)) %>%

mutate(euro\_cent\_asia=ifelse(wdi\_region=="Europe & Central Asia", 1, 0)) %>%

mutate(LA\_carrib=ifelse(wdi\_region=="Latin America & Caribbean", 1, 0)) %>%

mutate(MENA=ifelse(wdi\_region=="Middle East & North Africa", 1, 0)) %>%

mutate(N\_america=ifelse(wdi\_region=="North America", 1, 0)) %>%

mutate(S\_asia=ifelse(wdi\_region=="South Asia", 1, 0)) %>%

mutate(Sub\_africa=ifelse(wdi\_region=="Sub-Saharan Africa", 1, 0)) %>%

set\_variable\_labels(

e\_asia\_pacific="East Asia & Pacific",

euro\_cent\_asia="Europe & Central Asia",

LA\_carrib="Latin America & Caribbean",

MENA="Middle East & North Africa",

N\_america="North America",

S\_asia="South Asia",

Sub\_africa="Sub-Saharan Africa") %>%

select(-wdi\_region)

# 2. Regime data (V-Dem)

# 2.1. Reading in data.

url <- "https://github.com/vdeminstitute/vdemdata/raw/master/data/vdem.RData"

destfile <- tempfile(fileext = ".RData")

download.file(url, destfile, mode = "wb")

load(destfile)

rm(destfile, url)

# 2.2. Cleaning up data.

regime\_type <- vdem %>%

subset(select = c(country\_name, year, v2x\_regime)) %>%

rename(country = country\_name,

regime\_type = v2x\_regime) %>%

filter(year >= 1950)

regime\_type <- regime\_type %>%

left\_join(ccodes, by = c("country", "year")) %>% # NAs resulting from state-like actors, not full states.

subset(select = -c(country)) %>% # To prevent future duplicated columns.

drop\_na() # No duplicates either!

rm(vdem)

label(regime\_type$regime\_type) <- "0 = Closed autocracy, 1 = Electoral autocracy, 2 = Electoral democracy, 3 = Liberal Democracy"

# 2.3 Organizing variables for regression

regime\_type <- regime\_type %>%

mutate(

closed\_autocracy = ifelse(regime\_type == 0, 1, 0),

electoral\_autocracy = ifelse(regime\_type == 1, 1, 0),

electoral\_democracy = ifelse(regime\_type == 2, 1, 0),

liberal\_democracy = ifelse(regime\_type == 3, 1, 0)

)

# 2.4. Merging into data set.

base\_data <- base\_data %>%

left\_join(regime\_type, by = c("ccode", "year")) # Missing data simply is not updated by V-Dem, so I will not be dropping them.

rm(regime\_type)

# -------------------------- Social Data ------------------------------ #

# 1. Coup data (Powell & Thyne 2011).

# 1.1. Reading in data.

coup\_data <- read\_delim("http://www.uky.edu/~clthyn2/coup\_data/powell\_thyne\_coups\_final.txt",

delim = "\t", escape\_double = FALSE,

trim\_ws = TRUE)

# 1.2. Cleaning up data.

check <- coup\_data %>%

filter(ccode==lag(ccode) & year==lag(year) & month==lag(month))

#we have 6 cases of 2 coups in the same month:

#Haiti, 41, 04/1989; both failed

#Bolivia, 145, 05/1981; both failed

#Argentina, 160, 12/1975; both failed

#Sierra Leone, 451, 03/1967; second successful

#Togo, 461, 10/1991; both failed

#Sudan, 625, 12/1966; both failed

#Just need to make successful in Sierra Leone trump the failed one; all others are fine because both failed

table(coup\_data$coup)

coup\_data <- coup\_data %>%

mutate(coup=ifelse(ccode==451 & year==1967 & month==3, 2, coup))

table(coup\_data$coup) #looks good; proceed...

rm(check)

coup\_data <- coup\_data %>%

select(-ccode\_gw, -ccode\_polity, -day, -version) %>%

distinct() %>% #dropped the obs where 2 coups in same month, as above

mutate(coup\_attempt = 1) %>%

mutate(coup\_successful=ifelse(coup==2, 1, NA)) %>%

mutate(coup\_failed=ifelse(coup==1, 1, NA))

table(coup\_data$coup\_attempt)

table(coup\_data$coup, coup\_data$coup\_attempt)

table(coup\_data$coup, coup\_data$coup\_successful)

table(coup\_data$coup, coup\_data$coup\_failed)

#after merge, should have...

#487 attempts

#246 successful

#241 failed

#everything above looks good; proceed...

# 1.3. Merging into data set.

coup\_data <- coup\_data %>%

select(-country, -coup)

base\_data <- base\_data %>%

left\_join(coup\_data, by = c("year", "ccode", "month"))

base\_data <- base\_data %>%

mutate(coup\_attempt = ifelse(is.na(coup\_attempt) & year>=1950, 0, as.numeric(coup\_attempt))) %>%

mutate(coup\_successful = ifelse(is.na(coup\_successful) & year>=1950, 0, as.numeric(coup\_successful))) %>%

mutate(coup\_failed = ifelse(is.na(coup\_failed) & year>=1950, 0, as.numeric(coup\_failed)))

table(base\_data$coup\_attempt)

table(base\_data$coup\_successful)

table(base\_data$coup\_failed)

#we lost one successful coup because COW doesn't recognize Oman in 1970; we're good on this; proceed...

rm(coup\_data) # Keeping things clean

# 1.4. Set up non-coup months to deal with autocorrelation in the analyses

base\_data <- base\_data %>%

arrange(ccode, year, month) %>%

group\_by(ccode) %>%

mutate(sequence=row\_number())

base\_data <- base\_data %>%

btscs('coup\_attempt', 'year', 'ccode') %>%

rename(pce=spell\_time) %>%

mutate(pce2=pce\*pce) %>%

mutate(pce3=pce\*pce\*pce) %>%

set\_variable\_labels(

pce="Months since last coup",

pce2="Months^2",

pce3="Months^3") %>%

select(-sequence)

#For above, peace years are set up ignoring success/failed; go back and re-created these if you end up wanting to analyze success/failed instead of all attempts

# 2. Population data (World Bank Data Group 2024).

# 2.1. Reading in data.

url <- "https://extdataportal.worldbank.org/content/dam/sites/data/gender-data/data/data-gen/zip/indicator/population-number.zip"

zip\_file <- "population-data.zip"

download.file(url, zip\_file, mode = "wb")

unzip(zip\_file, exdir = "unzipped\_data")

world\_bank <- read.csv("unzipped\_data/Population (number).csv") # WB is missing data for 2024. This will update automatically after they add it.

rm(url, zip\_file)

unlink("population-data.zip")

unlink("unzipped\_data", recursive=TRUE)

# 2.2. Cleaning up data.

pop <- world\_bank %>%

filter(Indicator.Name == 'Population, total') %>%

subset(select = c(Country.Name, Year, Value)) %>%

rename(country = Country.Name,

year = Year,

population = Value) %>%

left\_join(ccodes, by = c("year", "country"))

check <- pop %>%

filter(if\_any(everything(), is.na))

table(check$country) #all regions or tiny countries; we're fine to drop NAs

pop <- pop %>%

drop\_na() %>%

select(-country)

rm(check)

check <- pop %>%

distinct() #no duplicates in pop data

check <- base\_data %>%

distinct() #no duplicates in base data

rm(check)

pop <- pop %>%

mutate(year=year+1) %>%

mutate(pop=log(population)) %>%

set\_variable\_labels(

pop="Total pop, WDI, log, t-1"

) %>%

select(-population)

# 2.3. Merging into data set.

base\_data <- base\_data %>%

ungroup() %>%

left\_join(pop, by = c("ccode", "year"))

rm(pop)

check <- base\_data %>%

filter(is.na(pop))

histogram(check$year) #looks fine but should be able to get pop for full sample from elsewhere

rm(check)

rm(world\_bank)

# 4. Median age data (World Bank Data Group 2024)

# 4.1. Reading in data.

url <- "https://ourworldindata.org/grapher/median-age.csv?v=1&csvType=full&useColumnShortNames=true"

median\_age <- read\_csv(url)

rm(url)

# 4.2. Cleaning up data.

median\_age <- median\_age %>%

subset(select = c(Entity, Year, median\_age\_\_sex\_all\_\_age\_all\_\_variant\_estimates, median\_age\_\_sex\_all\_\_age\_all\_\_variant\_medium)) %>%

rename(country = Entity,

year = Year,

popln\_median = median\_age\_\_sex\_all\_\_age\_all\_\_variant\_estimates,

popln\_median\_est = median\_age\_\_sex\_all\_\_age\_all\_\_variant\_medium) %>%

mutate(median\_age = ifelse(is.na(popln\_median), popln\_median\_est, popln\_median)) %>%

subset(select = -c(popln\_median, popln\_median\_est)) %>%

filter(year <= 2024)

median\_age <- median\_age %>%

mutate(year=year+1) %>% #just lagged

left\_join(ccodes, by = c("country", "year"))

check <- median\_age %>%

filter(is.na(ccode))

table(check$country) #fine; all regions or very small countries

rm(check)

median\_age <- median\_age %>%

drop\_na() %>% # NAs from non-country rows.

distinct() %>% # Random duplicates.

select(-country)

# 4.3. Merging into data set.

base\_data <- base\_data %>%

left\_join(median\_age, by = c("ccode", "year"))

rm(median\_age)

# ------------------------------------ Military data ------------------------------------ #

# 1.Military expenditure (% of GDP)

# 1.1 Loading data

url <- "https://api.worldbank.org/v2/en/indicator/MS.MIL.XPND.GD.ZS?downloadformat=excel"

destfile <- "MS\_MIL\_XPND\_GD.xls"

curl::curl\_download(url, destfile)

milex <- read\_excel(destfile, skip = 3)

rm(destfile, url)

#1.2 Reshaping data

milex <- milex %>%

rename( "country" = `Country Name`) %>%

subset(select = -c(`Indicator Name`, `Indicator Code`, `Country Code`)) #removing things we don't want

milex <- milex %>%

pivot\_longer(

cols = -c(country), # Keep country-related columns fixed

names\_to = "year",

values\_to = "military expenditure"

) %>%

mutate(year = as.integer(year)) # Convert Year to integer

#merge to base

milex <- milex %>%

rename(milex='military expenditure') %>%

set\_variable\_labels(milex="Milit expenditure, % of GDP") %>%

mutate(year=year+1) #lagged

milex <- milex %>%

left\_join(ccodes, by=c("country", "year")) %>%

select(-country) %>%

distinct()

base\_data <- base\_data %>%

left\_join(milex, by=c("ccode", "year"))

rm(milex)

# 2.Armed forces personnel (total)

# 2.1 Loading data

url <- "https://api.worldbank.org/v2/en/indicator/MS.MIL.TOTL.P1?downloadformat=excel"

destfile <- "MS\_MIL\_TOTL.xls"

curl::curl\_download(url, destfile)

milper <- read\_excel(destfile, skip = 3)

rm(destfile, url)

# 2.2 Reshaping data; cleaning a bit

milper <- milper %>%

rename( "country" = `Country Name`) %>%

subset(select = -c(`Indicator Name`, `Indicator Code`, `Country Code`)) #removing things we don't want

milper <- milper %>%

pivot\_longer(

cols = -c(country), # Keep country-related columns fixed

names\_to = "year",

values\_to = "milper"

) %>%

mutate(year = as.integer(year)) # Convert Year to integer

milper <- milper %>%

drop\_na() %>%

mutate(milper=log(milper+1)) %>% #just logged

mutate(year=year+1) %>% #just lagged

set\_variable\_labels(milper="milit personell; logged; t-1")

# 3 Merging

milper <- milper %>%

left\_join(ccodes, by=c("country", "year")) %>%

drop\_na() %>%

select(-country)

base\_data <- base\_data %>%

left\_join(milper, by=c("ccode", "year"))

rm(milper)

###############################################################################################

#Checked through above and ready to produce .csv and upload to github

write.csv(base\_data, gzfile("base\_data.csv.gz"), row.names = FALSE)

#Now push push the file that was just written to the working directory to github

###############################################################################################

#### Filling in NAs below, committed on 3/7 ####

# Source the interpolation functions script

source("interpolation\_functions.R")

# Load the dataset

url <- "https://raw.githubusercontent.com/thynec/CoupCats/refs/heads/main/base\_data.csv"

df <- read.csv(url)

# Check for missing values

values <- check\_missing\_values(df)

# Fill missing values using forward/backward fill

df <- fill\_missing\_values(df, values)

# Interpolate missing values

for (val in values) {

df <- interpolate\_missing\_values(df, val, max\_gap = 36, method = 'spline', order = 3)

}

# Check remaining missing values

cat("\nFinal missing values check:\n")

for (col in c(values, paste0(values, '\_interpolated'))) {

if (col %in% colnames(df)) {

na\_count <- sum(is.na(df[[col]]))

if (na\_count > 0) {

total <- nrow(df)

pct <- na\_count / total \* 100

cat(sprintf("%s has %d / %d rows (%.2f%%) missing values\n", col, na\_count, total, pct))

}

}

}

#### End of Commit ####

###############################################################################################

#Below is stuff we probably won't use but keeping it here just in case...

###############################################################################################

# 3. Age population data (World Bank Data Group 2024).

# 3.1. Reading in data.

#age\_popln <- world\_bank

#%>%

# 3.2. Cleaning up data.

# filter(Indicator.Name %in% c('Population ages 0-14, total',

# 'Population ages 15-64, total',

# 'Population ages 65 and above, total')) %>%

# subset(select = c(Country.Name, Year, Value, Disaggregation)) %>%

# rename(country = Country.Name,

# year = Year,

# age\_tot = Value,

# type = Disaggregation)

#age\_popln <- age\_popln %>%

# left\_join(ccodes, by = c("country", "year"), relationship = "many-to-many") %>%

# drop\_na() %>% # NAs from non-country rows.

# distinct() # Random duplicates.

#age\_popln <- age\_popln %>%

# pivot\_wider(names\_from = type, values\_from = age\_tot, values\_fn = first) %>%

# rename(age0\_14 = `total, 0-14`,

# age15\_64 = `total, 15-64`,

# age65plus = `total, 65+`)

# 3.3. Merging into data set.

#emma\_data <- emma\_data %>% # No 2024 data.

# left\_join(age\_popln, by = c("country", "year", "ccode"), relationship = "many-to-many") %>%

# distinct() # Random duplicates.

#rm(world\_bank, age\_popln)

# 5. Age expectancy data.

# 5.1. Reading in data.

#url <- "https://srhdpeuwpubsa.blob.core.windows.net/whdh/DATADOT/INDICATOR/C64284D\_ALL\_LATEST.csv"

#age\_expectancy <- read\_csv(url)

#rm(url)

# 5.2. Cleaning up data.

#age\_expectancy <- age\_expectancy %>%

# subset(select = c(DIM\_TIME, GEO\_NAME\_SHORT, DIM\_SEX, AMOUNT\_N)) %>%

# filter(DIM\_SEX == 'TOTAL') %>%

# rename(country = GEO\_NAME\_SHORT,

# year = DIM\_TIME,

# age\_expectancy = AMOUNT\_N) %>%

# subset(select = -c(DIM\_SEX))

#age\_expectancy <- age\_expectancy %>%

# left\_join(ccodes, by = c("country", "year")) %>% # Missing 'Côte d'Ivoire', 'Türkiye', 'Zambia', 'Zimbabwe'

# drop\_na() %>% # NAs from non-country rows.

# distinct() # Random duplicates from Yemen.

# 5.3. Merging into data set.

#emma\_data <- emma\_data %>%

# left\_join(age\_expectancy, by = c("country", "year", "ccode"), relationship = "many-to-many")

#rm(age\_expectancy)