Comp2_panel_wrangling.R

sevastiansanchez

2025-08-11

```r
# set working directory
setwd("~/Documents/GitHub/QMSS_Thesis_Sanchez")

#load libraries/packages
source("packages.R")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Loading required package: carData
##
##
## Attaching package: 'car'
##
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
##
## The following object is masked from 'package:purrr':
##
##     some
##
##
## Loading required package: usethis
##
##
## Attaching package: 'ERT'
##
##
## The following objects are masked from 'package:vdemdata':
##
##     codebook, vdem
##
##
```

```
##
## Please cite as:
##
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
##
##
##
## Attaching package: 'scales'
##
##
## The following object is masked from 'package:purrr':
##
##     discard
##
##
## The following object is masked from 'package:readr':
##
##     col_factor
##
##
##
## Attaching package: 'kableExtra'
##
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
##
##
##
## Attaching package: 'mice'
##
##
## The following object is masked from 'package:stats':
##
##     filter
##
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
##
##
## Loading required package: MASS
##
##
## Attaching package: 'MASS'
##
##
## The following object is masked from 'package:dplyr':
##
```

```
##     select
##
##
##
## Attaching package: 'plm'
##
##
## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
##
##
##
## Attaching package: 'patchwork'
##
##
## The following object is masked from 'package:MASS':
##
##     area
##
##
##
## Attaching package: 'reshape2'
##
##
## The following object is masked from 'package:tidyr':
##
##     smiths
##
##
##
## Attaching package: 'jsonlite'
##
##
## The following object is masked from 'package:purrr':
##
##     flatten
##
##
## Loading required package: zoo
##
##
## Attaching package: 'zoo'
##
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## Loading required package: Matrix
##
##
## Attaching package: 'Matrix'
```

```
##
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
## Loading required package: mvtnorm
##
## mediation: Causal Mediation Analysis
## Version: 4.5.0
##
##
##
## Attaching package: 'plotly'
##
##
## The following object is masked from 'package:MASS':
##
##     select
##
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
##
## The following object is masked from 'package:stats':
##
##     filter
##
##
## The following object is masked from 'package:graphics':
##
##     layout
##
##
##
## Attaching package: 'ggdag'
##
##
## The following object is masked from 'package:stats':
##
##     filter
```

```r
#load function
#source("df_years2.0_Function.R")

#load df_years() function: 2015-present
#all_data <- df_years2.0(2004, 2023)

#load data
all_data <- read_csv("data/Main CSV Outputs/merged_final_df.csv")
```

```
## Rows: 3340 Columns: 46
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): country_name, country_code, income_level, income_level_lab
## dbl (42): year, sdg_overall, spi_comp, sci_overall, di_score, regime_type_2,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# selecting vars
panel_data <- all_data %>%
  dplyr::select(country_name, country_code, year, sdg_overall, spi_comp, sci_overall,
                di_score, elect_dem, aut_ep, dem_ep, regime_type_4, regch_event, log_gdppc,
                income_level, goal1:goal17, p1_use, p2_services, p3_products, p4_sources,
                p5_infra) %>%
  arrange(country_code, year) %>%  # Critical for correct lagging
  filter(year >= 2016)

# Transforming variables: Centering >> Lagging >> Squaring & Cubing Terms (for polynomial terms)
panel_data <- panel_data %>%
  group_by(country_code) %>%
  arrange(year) %>%  # Ensure data is sorted by year within each country
  mutate(
    # Transforming SPI
    cen_spi_comp = spi_comp - mean(spi_comp, na.rm = TRUE),
    cen_spi_comp_lag1 = dplyr::lag(cen_spi_comp, n = 1),
    cen_spi_comp_lag1_sq = cen_spi_comp_lag1^2,
    cen_spi_comp_lag1_cub = cen_spi_comp_lag1^3,
    cen_spi_comp_lag2 = dplyr::lag(cen_spi_comp, n = 2),
    cen_spi_comp_lag2_sq = cen_spi_comp_lag2^2,
    cen_spi_comp_lag2_cub = cen_spi_comp_lag2^3,

    # Transforming DI
    cen_di_score = di_score - mean(di_score, na.rm = TRUE),
    cen_di_score_lag1 = dplyr::lag(cen_di_score, n = 1),
    cen_di_score_lag1_sq = cen_di_score_lag1^2,
    cen_di_score_lag1_cub = cen_di_score_lag1^3,
    cen_di_score_lag2 = dplyr::lag(cen_di_score, n = 2),
    cen_di_score_lag2_sq = cen_di_score_lag2^2,
    cen_di_score_lag2_cub = cen_di_score_lag2^3,

    # Transforming log GDP per capita
    cen_log_gdppc = log_gdppc - mean(log_gdppc, na.rm = TRUE),
    cen_log_gdppc_sq = cen_log_gdppc^2,
    cen_log_gdppc_cub = cen_log_gdppc^3
  ) %>%
  ungroup()

# Creating first and second order lags for di_score, spi_comp, and log_gdppc
panel_data <- panel_data %>%
  group_by(country_code) %>%
  arrange(year) %>%  # Ensure data is sorted by year within each country
  mutate(
    di_score_lag1 = dplyr::lag(di_score, n = 1),
    di_score_lag2 = dplyr::lag(di_score, n = 2),
    spi_comp_lag1 = dplyr::lag(spi_comp, n = 1),
    spi_comp_lag2 = dplyr::lag(spi_comp, n = 2),
    log_gdppc_lag1 = dplyr::lag(log_gdppc, n = 1),
    log_gdppc_lag2 = dplyr::lag(log_gdppc, n = 2)
  ) %>%
  ungroup()

##### GNI INCOME LEVEL VARIABLES #####
# Recoding income_level, split income_level into dummy variables using case_when()
# Everything on the left of ~ is the condition, and everything on the right
# Is the value to return if the condition is true
panel_data <- panel_data %>%
  mutate(income_level_recoded = case_when(
    income_level == "L" ~ 0, # Low-Income
    income_level == "LM" ~ 1, # Lower-Middle-Income
    income_level == "UM" ~ 2, # Upper-Middle-Income
    income_level == "H" ~ 3, # High-Income
    TRUE ~ NA_integer_  # Handle any other cases
  )) %>%
  mutate(income_level_recoded = as.factor(income_level_recoded)) %>%

  #### REGIME TYPE VARIABLES ####
  # factorizing regime_type_4 (RoW based): 0 = Autocracy; 1 = Democracy
  mutate(regime_type_4 = as.factor(regime_type_4)) %>%
  # creating two variables for autocracy and democracy dummies (RoW based)
  mutate(
    autocracy = case_when(
      regime_type_4 == 0 ~ 1, # Autocracy
      regime_type_4 == 1 ~ 1, # Autocracy
      regime_type_4 == 2 ~ 0, # Democracy
      regime_type_4 == 3 ~ 0, # Democracy
      TRUE ~ NA_integer_ # Handle any other cases
    ),
    democracy = case_when(
      regime_type_4 == 0 ~ 0, # Autocracy
      regime_type_4 == 1 ~ 0, # Autocracy
      regime_type_4 == 2 ~ 1, # Democracy
      regime_type_4 == 3 ~ 1, # Democracy
      TRUE ~ NA_integer_ # Handle any other cases
    )
  ) %>%
  # Creating a new regime type var (di_score)
  mutate(
    di_reg_type_2 = case_when(
      di_score < 5 ~ 0,  # Autocracy
      di_score >= 5 ~ 1,  # Democracy
      TRUE ~ NA_integer_
  )) %>%
  # Convert to factors
  mutate(autocracy = as.factor(autocracy), # autocracy dummy
         democracy = as.factor(democracy), # democracy dummy
         di_reg_type_2 = as.factor(di_reg_type_2)) # regime type dummy (di based)
```

```
#### REGIME CHANGE VARIABLES ####
panel_data <- panel_data %>%
  # factorize variables first
  mutate(
    aut_ep = as.factor(aut_ep), # autocratization episode
    dem_ep = as.factor(dem_ep),  # democratization episode
    regch_event = as.factor(regch_event) # regime change event
  ) %>%
  # Group by country to check for any event
  group_by(country_code) %>%
  # has atleast 1 autocratization episode
  mutate(has_aut_ep = case_when(any(aut_ep == 1, na.rm = TRUE) ~ 1, TRUE ~ 0)) %>%
  # has atleast 1 democratization episode
  mutate(has_dem_ep = case_when(any(dem_ep == 1, na.rm = TRUE) ~ 1, TRUE ~ 0)) %>%
  # has neither autocratization nor democratization episodes
  mutate(has_neither = case_when(!any(aut_ep == 1 | dem_ep == 1, na.rm = TRUE) ~ 1, TRUE ~ 0)) %>%
  # two new variable for the sum of aut_ep and dem_ep episodes
  mutate(total_aut_ep = sum(as.numeric(as.character(aut_ep)), na.rm = TRUE)) %>%
  mutate(total_dem_ep = sum(as.numeric(as.character(dem_ep)), na.rm = TRUE)) %>%

  ## Complete Change Variables ##
  # change from autocracy >> democracy
  mutate(democratized = case_when(any(regch_event == 1, na.rm = TRUE) ~ 1, TRUE ~ 0)) %>%
  # change from democracy >> autocracy
  mutate(autocratized = case_when(any(regch_event == -1, na.rm = TRUE) ~ 1, TRUE ~ 0)) %>%
  # stable regime (no change)
  mutate(stable = case_when(!any(regch_event == 1 | regch_event == -1, na.rm = TRUE) ~ 1, TRUE ~ 0)) %>
  ungroup() %>%
  # Convert to factors
  mutate(
    aut_ep = as.factor(aut_ep), # autocratization episode
    dem_ep = as.factor(dem_ep), # democratization episode
    has_aut_ep = as.factor(has_aut_ep), # has autocratization episode
    has_dem_ep = as.factor(has_dem_ep), # has democratization episode
    has_neither = as.factor(has_neither), # has neither autocratization nor democratization episodes
    democratized = as.factor(democratized), # democratized
    autocratized = as.factor(autocratized), # autocratized
    stable = as.factor(stable) # stable regime
  )

# reorder columns
panel_data <- panel_data %>%
  select(country_name, country_code, year, sdg_overall, spi_comp, sci_overall, di_score, di_reg_type_2,
         aut_ep, dem_ep, has_aut_ep, has_dem_ep, total_aut_ep, total_dem_ep, has_neither, regime_type_4
         log_gdppc, income_level, income_level_recoded,
         goal1:goal17, p1_use, p2_services, p3_products, p4_sources, p5_infra, everything())


#### YEAR TO YEAR LAGS FOR FD MODELS (NEW DF) ####
fd_data <- panel_data %>%
  select(country_code, year, sdg_overall, di_score, spi_comp, log_gdppc, income_level, aut_ep,
         dem_ep, income_level_recoded, regch_event, di_score_lag1, di_score_lag2, spi_comp_lag1,
         spi_comp_lag2, log_gdppc_lag1, log_gdppc_lag2) %>%
```

```r
  filter(!is.na(di_score) & !is.na(spi_comp) | !is.na(spi_comp) & !is.na(sdg_overall)) %>%
  group_by(country_code) %>%
  arrange(year) %>%  # Ensure data is sorted by year within each country
  mutate(
    # first differences for selected variables
    sdg_diff = sdg_overall - dplyr::lag(sdg_overall, n=1),
    di_diff = di_score - di_score_lag1,
    spi_diff = spi_comp - spi_comp_lag1,
    log_gdppc_diff = log_gdppc - log_gdppc_lag1,
    # lagged first differences
    di_diff_lag1 = dplyr::lag(di_diff, n=1),
    di_diff_lag2 = dplyr::lag(di_diff, n=2),
    spi_diff_lag1 = dplyr::lag(spi_diff, n=1),
    spi_diff_lag2 = dplyr::lag(spi_diff, n=2),
    log_gdppc_diff_lag1 = dplyr::lag(log_gdppc_diff, n=1),
    log_gdppc_diff_lag2 = dplyr::lag(log_gdppc_diff, n=2)
  ) %>%
  ungroup()

# View(panel_data)
# View(fd_data)
```