KDI Foundational DB

Harmonizing Macro-Level Democratic Indicators

Moosa Ghaeb

August 2, 2025

Project Overview

This document contains the R code for a data pipeline project designed to harmonize several key datasets in comparative political science. The project's objective is to ingest raw data from The World Bank, the V-Dem Institute, and the Quality of Government (QoG) Institute, process and link them deterministically, and load the final, analysis-ready dataset into a PostgreSQL database.

The pipeline is structured across several scripts, each with a distinct purpose, demonstrating a modular and reproducible workflow.

Script 1: Data Ingestion

`R/01_ingest_data.R`

This script handles the "Extract" phase. It downloads all necessary raw data from the three sources and saves them locally.

```r
# ---------------------------------------------------------------------------
# SCRIPT: 01_ingest_and_assess_data.R
# PROJECT: KDI Foundational Database
#
# PURPOSE: 1. Load necessary libraries.
#          2. Ingest raw data from three sources.
#
# ---------------------------------------------------------------------------


if (!require("wbstats")) install.packages("wbstats")
if (!require("dplyr")) install.packages("dplyr")
if (!require("haven")) install.packages("haven")

library(wbstats)
library(dplyr)
library(haven)



dir.create("data/vdem", showWarnings = FALSE, recursive = TRUE)
dir.create("data/qog", showWarnings = FALSE, recursive = TRUE)

```

```r

# World Bank Data (via API)
wb_raw_data <- wb_data(
  indicator = "NY.GDP.PCAP.KD",
  start_date = 1990,
  end_date = 2023
)


# V-Dem Data
vdem_url <- "https://www.v-dem.net/media/datasets/V-Dem-CY-Core-v15_dta.zip"
vdem_zip_path <- "data/vdem/V-Dem-CY-Core-v15.zip"
vdem_dta_path <- "data/vdem/V-Dem-CY-Core-v15.dta"

if (!file.exists(vdem_zip_path)) {
  download.file(vdem_url, destfile = vdem_zip_path, mode = "wb")
}
if (!file.exists(vdem_dta_path)) {
  unzip(vdem_zip_path, exdir = "data/vdem")
}
vdem_raw_data <- read_dta(vdem_dta_path)


# QoG Data
qog_url <- "https://www.qogdata.pol.gu.se/data/qog_std_ts_jan25.csv"
qog_csv_path <- "data/qog/qog_std_ts_jan25.csv"

if (!file.exists(qog_csv_path)) {
  download.file(qog_url, destfile = qog_csv_path, mode = "wb")
}
qog_raw_data <- read.csv(qog_csv_path)
```

Script 2: Data Transformation and Merging

R/02_transform_and_merge.R

This script performs the "Transform" phase. It cleans each raw dataset, filters for the relevant time period (1990+), standardizes column names, and merges the three sources into a single, coherent data frame using a key-based full join.

```r
# ---------------------------------------------------------------------------
# SCRIPT: 02_transform_and_merge.R
# PROJECT: KDI Foundational Database
#
# PURPOSE: 1. Cleans and standardizes each of the three datasets.
#          2. Filters each dataset to the relevant time period (1990+).
#          3. Merges the clean datasets into a single master data frame.
#
# ---------------------------------------------------------------------------


# world bank
wb_clean <- wb_raw_data %>%
  filter(date >= 1990) %>%
  select(iso3c, country, date, NY.GDP.PCAP.KD) %>%
  rename(
    country_iso3 = iso3c,
    country_name = country,
    year = date,
    wb_gdp_pc = NY.GDP.PCAP.KD
  )

```

```

# vdem
vdem_clean <- vdem_raw_data %>%
  filter(year >= 1990) %>%
  select(country_text_id, country_name, year, v2x_polyarchy) %>%
  rename(
    country_iso3 = country_text_id,
    vdem_polyarchy = v2x_polyarchy
  )


# qog
qog_clean <- qog_raw_data %>%
  filter(year >= 1990) %>%
  select(ccodealp, cname, year, bci_bci) %>%
  rename(
    country_iso3 = ccodealp,
    country_name = cname,
    qog_bci = bci_bci
  )



# join
merged_data_temp <- vdem_clean %>%
  full_join(qog_clean, by = c("country_iso3", "year")) %>%
  full_join(wb_clean, by = c("country_iso3", "year"))


merged_data <- merged_data_temp %>%
  mutate(country_name = coalesce(country_name.x, country_name.y, country_name))
    %>%
  select(country_iso3, country_name, year, vdem_polyarchy, qog_bci, wb_gdp_pc)
```

Script 3: Load Data to PostgreSQL

R/03_load_to_db.R

This script handles the "Load" phase. It connects to a PostgreSQL database and writes the final, merged data frame to a table, making it available for analysis.

```
# -------------------------------------------------------------------------
# SCRIPT: 03_load_to_db.R
# PROJECT: KDI Foundational Database
#
# PURPOSE: This script takes the final, merged dataframe from the R
#          environment and writes it to the 'harmonized_data' table
#          in the PostgreSQL database.
#
# -------------------------------------------------------------------------


if (!require("DBI")) install.packages("DBI")
if (!require("RPostgres")) install.packages("RPostgres")

library(DBI)
library(RPostgres)


# Establish the connection using credentials stored in .Renviron
con <- dbConnect(RPostgres::Postgres(),
                 dbname = "kdi_db",
                 host = "localhost",
                 port = 5432,
                 user = "postgres",
                 password = Sys.getenv("DB_PASSWORD")
)


dbWriteTable(con,
             "harmonized_data",
             merged_data,
             overwrite = TRUE,
             row.names = FALSE)

dbDisconnect(con)
```

Script 4: Methodological Extension - Imputation

R/04_imputation_example.R

This script demonstrates a methodological extension for handling missing data. It uses linear interpolation to impute missing values in the time-series data for GDP and creates a visualization to illustrate the result.

```
# ---------------------------------------------------------------------------
# SCRIPT: 04_imputation_example.R
#
# PURPOSE: linear interpolation.
# ---------------------------------------------------------------------------


if (!require("dplyr")) install.packages("dplyr")
if (!require("imputeTS")) install.packages("imputeTS")
if (!require("ggplot2")) install.packages("ggplot2")

library(dplyr)
library(imputeTS)
library(ggplot2)



imputed_data <- merged_data %>%
  group_by(country_iso3) %>%
  arrange(year) %>%
  mutate(
    imputed_gdp = if (sum(!is.na(wb_gdp_pc)) >= 2) {
      na_interpolation(wb_gdp_pc)
    } else {
      wb_gdp_pc
    }
  ) %>%
  ungroup()



imputed_data %>%
  filter(country_iso3 == "POL") %>%
  ggplot(aes(x = year)) +
  geom_line(aes(y = imputed_gdp), color = "dodgerblue", linewidth = 1) +
  geom_point(aes(y = wb_gdp_pc), color = "red", size = 2.5) +
  labs(
    title = "GDP per Capita for Poland (Imputed vs. Original)",
```

```r
    subtitle = "Red points are original data. The blue line shows the imputed time-series.",
    x = "Year",
    y = "GDP per Capita (Constant 2015 US$)",
    caption = "Source: World Bank data with linear interpolation for missing values."
  ) +
  theme_minimal()
```