

```

sink("console_data_done_ver75.txt")

# y1lg & y2lg
# One-way FE/RE + Hausman test

# Load packages ----
# install.packages(c("fixest", "plm", "lmtest", "sandwich", "readxl", "car",
"performance"))
library(fixest)
library(plm)
library(lmtest)
library(sandwich)
library(readxl)
library(car)
library(performance)
library(dplyr)

# Read data ----
# setwd("/Users/mynhie/Desktop/Thesis/Fall 2025 version/data done/R")
panel_data <- read_excel("Panel Data finalfinalfinalfinalfinal.xlsx")

# Prepare variables ----
panel_data$Subredit <- as.factor(panel_data$Subredit)
panel_data$Time <- as.factor(panel_data$Time)

panel_data <- panel_data %>%
  mutate(across(
    starts_with("log_"),
    ~ ifelse(is.na(.x), 0, .x)
  ))
print(summary(panel_data$y1lg))
print(sum(is.na(panel_data$y1lg)))
print(sum(is.infinite(panel_data$y1lg)))

pdata <- pdata.frame(panel_data, index = c("Subredit", "Time"))

vars <- c("y1", "y2", "y1lg", "y2lg")

descriptives <- data.frame(
  Variable = vars,
  Mean = sapply(panel_data[vars], mean, na.rm = TRUE),
  SD = sapply(panel_data[vars], sd, na.rm = TRUE),
  Min = sapply(panel_data[vars], min, na.rm = TRUE),
  Max = sapply(panel_data[vars], max, na.rm = TRUE)
)

descriptives_clean <- data.frame(
  lapply(descriptives, function(x) {
    if(is.numeric(x)) format(x, scientific = FALSE) else x
  }),
  check.names = FALSE
)
print(descriptives_clean)

cat("===== PANEL ECONOMETRICS RESULTS =====\n\n")

# ONE-WAY (Subreddit FE/RE) + HAUSMAN TEST

cat("\n===== \nONE-WAY MODELS \n===== \n")

## ----- Model 1: y1lg -----
cat("\n===== FE MODEL: y1lg =====\n")
fe_y1_oneway <- plm(y1lg ~ R1 + R1:R2 + R1:R3
  + D3 + lag(D4,1)
  + S1
  + Control_NoSub + Control_LifeTime + Control_Protest,
  data = pdata, model = "within")
print(summary(fe_y1_oneway))

```

```

cat("\n===== RE MODEL: y1lg =====\n")
re_y1_owenay <- plm(y1lg ~ R1 + R1:R2 + R1:R3
                    + D3 + lag(D4,1)
                    + S1
                    + Control_NoSub + Control_LifeTime + Control_Protest,
                    data = pdata, model = "random")
print(summary(re_y1_owenay))

cat("\n===== HAUSMAN TEST (y1lg) =====\n")
print(phptest(fe_y1_owenay, re_y1_owenay))

## ----- Model 2: y2lg -----
cat("\n===== FE MODEL: y2lg =====\n")
fe_y2_owenay <- plm(y2lg ~ R1 + R1:R2 + R1:R3
                    + D3 + lag(D4,1)
                    + S1
                    + Control_NoSub + Control_LifeTime + Control_Protest,
                    data = pdata, model = "within")
print(summary(fe_y2_owenay))

cat("\n===== RE MODEL: y2lg =====\n")
re_y2_owenay <- plm(y2lg ~ R1 + R1:R2 + R1:R3
                    + D3 + lag(D4,1)
                    + S1
                    + Control_NoSub + Control_LifeTime + Control_Protest,
                    data = pdata, model = "random")
print(summary(re_y2_owenay))

cat("\n===== HAUSMAN TEST (y2lg) =====\n")
print(phptest(fe_y2_owenay, re_y2_owenay))

# ---- POOLABILITY TEST ----
cat("\n===== POOLABILITY TEST =====\n")
# Pooled OLS model
pooled_y1 <- plm(
  y1lg ~ R1 + R1:R2 + R1:R3
  + D3 + lag(D4,1)
  + S1
  + Control_NoSub + Control_LifeTime + Control_Protest,
  data = pdata,
  model = "pooling"
)
pooled_y2 <- plm(
  y2lg ~ R1 + R1:R2 + R1:R3
  + D3 + lag(D4,1)
  + S1
  + Control_NoSub + Control_LifeTime + Control_Protest,
  data = pdata,
  model = "pooling"
)

# Poolability test
pooltest_y1 <- pooltest(pooled_y1, fe_y1_owenay)
pooltest_y2 <- pooltest(pooled_y2, fe_y2_owenay)
print(pooltest_y1)
print(pooltest_y2)

cat("\n===== TIME DUMMY TEST =====\n")
fe_time_model <- plm(
  y1lg ~ R1 + R1:R2 + R1:R3
  + D3 + lag(D4,1) + S1 +
  Control_NoSub + Control_LifeTime + Control_Protest +
  factor(Time),
  data = pdata,
  model = "within"
)

```

```

)
time_coef <- grep("factor\\(Time\\)", names(coef(fe_time_model)), value = TRUE)
time_coef
print(linearHypothesis(fe_time_model, time_coef))

# DIAGNOSTIC TESTS

cat("\n===== \nDIAGNOSTIC TESTS ONE-WAY \n===== \n")

# Multicollinearity
cat("\n===== Multicollinearity ===== \n")
vif_model_y1 <- lm(
  y1lg ~ R1 + R1:R2 + R1:R3 + D3 + lag(D4,1) + S1 +
  Control_NoSub + Control_LifeTime + Control_Protest,
  data = panel_data
)
vif_model_y2 <- lm(
  y2lg ~ R1 + R1:R2 + R1:R3 + D3 + lag(D4,1) + S1 +
  Control_NoSub + Control_LifeTime + Control_Protest,
  data = panel_data
)

print(vif(vif_model_y1))
print(vif(vif_model_y2))

# Heteroskedasticity
cat("\n===== BREUSCH-PAGAN TEST Heteroskedasticity ===== \n")
print(bptest(fe_y1_oweway))
print(bptest(re_y2_oweway))

# Serial correlation
cat("\n===== WOOLDRIDGE TEST Serial correlation ===== \n")
print(pwtest(fe_y1_oweway))
print(pwtest(re_y2_oweway))

# Cross-sectional dependence
cat("\n===== PESARAN CD TEST Cross-sectional dependence ===== \n")
print(pcdtest(fe_y1_oweway))
print(pcdtest(re_y2_oweway))

cat("\n===== DRISCOLL-KRAAY ===== \n")

# 2. Driscoll-Kraay standard errors
dk_y1 <- vcovSCC(fe_y1_oweway, type = "HC3", maxlag = 4)
dk_y2 <- vcovSCC(re_y2_oweway, type = "HC3", maxlag = 4)

# 3. Summary with DK standard errors
print(coeftest(fe_y1_oweway, vcov = dk_y1))
print(coeftest(re_y2_oweway, vcov = dk_y2))

sink()

```