Data Manegement&Analysis Final Project

Replication and Extention for Acemoglu, Naidu, Restrepo and Robinson (2019)

(Name:) Shoya Abe (University ID:) 31B24001
(Name:) Honoka Ohtani (University ID:) 31B24002

(Submission Due:) 2024/02/06

Contents

0.1 Setup . . . 2

1 About this Report 2

1.1 Project Type . . . 2
1.2 Summary of the Paper . . . 2
1.2.1 What the problem is . . . 2
1.2.2 Why it is important . . . 2
1.2.3 How you solve the problem . . . 2
1.2.4 What we find . . . 2
1.3 Data . . . 2
1.4 Emprical Methods . . . 3
1.4.1 Event Study (Figure.1) . . . 3
1.4.2 Dynamic Panel Data Model (Table.2) . . . 3

2 Replication 3

2.1 Figure.1 . . . 3
2.1.1 Preprocessing . . . 3
2.1.2 Estimation . . . 4
2.1.3 Plot . . . 6
2.2 Table.1 . . . 7
2.3 Table.2 . . . 7
2.4 Figure.2 . . . 7
2.5 Table.3 . . . 7
2.6 Table.4 . . . 7
2.7 Table.5 . . . 7

3 Extention 7

0.1 Setup

```
pacman::p_load(
  rmdformats,
  knitr,
  tinytex,
  haven,
  tidyverse,
  fastDummies,
  skimr
)

## Global options
options(max.print="75")
opts_chunk$set(fig.align="center",
               echo=TRUE,
               cache=TRUE,
               prompt=FALSE,
               tidy=TRUE,
               comment=NA,
               message=FALSE,
               warning=FALSE)
opts_knit$set(width=75)
theme_set(theme_bw(base_size = 14))
```

1 About this Report

1.1 Project Type

1.2 Summary of the Paper

1.2.1 What the problem is

1.2.2 Why it is important

1.2.3 How you solve the problem

1.2.4 What we find

1.3 Data

```
data <- read_dta("data/raw/DDCGdata_final.dta")

# Define the function to summarize the data
summarize_data <- function(data, n = 10) {
    cat("Sample size (number of rows):", nrow(data), "\n")
    cat("Number of variables (columns):", ncol(data), "\n")
    cat("Variable names (first", n, "names):\n")
    print(head(colnames(data), n))
```

```
}

summarize_data(data)
```

```
Sample size (number of rows): 9384
Number of variables (columns): 1177
Variable names (first 10 names):
 [1] "country_name"                "wbcode"
 [3] "year"                        "gdppercapitaconstant2000us"
 [5] "lp_bl"                       "ls_bl"
 [7] "lh_bl"                       "taxratio"
 [9] "region"                      "wbcode2"
```

1.4 Emprical Methods

1.4.1 Event Study (Figure.1)

1.4.2 Dynamic Panel Data Model (Table.2)

2 Replication

2.1 Figure.1

2.1.1 Preprocessing

```
# Rename '_ID' to 'id', sort by year within each id, then ungroup
data_f1 <- data %>%
    rename(id = "_ID") %>%
    group_by(id) %>%
    arrange(year) %>%
    ungroup()

# For each id, create a lagged democracy indicator and define the transition: -
# Transition = 1 if it goes from 0 to 1 (non-democracy to democracy) -
# Transition = 0 if it remains at 0 (non-democracy) - Otherwise, transition is
# set to NA
data_f1 <- data_f1 %>%
    group_by(id) %>%
    arrange(year) %>%
    mutate(prev_dem = lag(dem, 1)) %>%
    ungroup() %>%
    mutate(transition = case_when(dem == 1 & prev_dem == 0 ~ 1, dem == 0 & prev_dem ==
        0 ~ 0, TRUE ~ NA_real_))

# Compute lags for 'y' (lags 1 through 4) within each id and filter out rows
# with any missing lag values
data_f1 <- data_f1 %>%
    group_by(id) %>%
    arrange(year) %>%
    mutate(lag1 = lag(y, 1), lag2 = lag(y, 2), lag3 = lag(y, 3), lag4 = lag(y, 4)) %>%
```

```r
    ungroup() %>%
    filter(!is.na(lag1) & !is.na(lag2) & !is.na(lag3) & !is.na(lag4))

# Calculate GDP differences for past periods: For t from -15 to -2, create
# columns 'gdpDiff_m[abs(t)]' as the difference between the lagged 'y' (by
# abs(t)) and lag1
for (t in -15:-2) {
    col_name <- paste0("gdpDiff_m", abs(t))
    data_f1 <- data_f1 %>%
        group_by(id) %>%
        arrange(year) %>%
        mutate(!!col_name := lag(y, abs(t)) - lag1) %>%
        ungroup()
}

# Set the GDP difference for the immediate past period (m1) to 0
data_f1 <- data_f1 %>%
    mutate(gdpDiff_m1 = 0)

# Compute the GDP difference for the current period (0) as the difference
# between 'y' and lag1
data_f1 <- data_f1 %>%
    group_by(id) %>%
    arrange(year) %>%
    mutate(gdpDiff_0 = y - lag1) %>%
    ungroup()

# Calculate GDP differences for future periods: For t from 1 to 30, create
# columns 'gdpDiff_p[t]' as the difference between the lead of 'y' (by t) and
# lag1
for (t in 1:30) {
    col_name <- paste0("gdpDiff_p", t)
    data_f1 <- data_f1 %>%
        group_by(id) %>%
        arrange(year) %>%
        mutate(!!col_name := lead(y, t) - lag1) %>%
        ungroup()
}

# Keep only rows where the transition value is not missing
data_f1 <- data_f1 %>%
    filter(!is.na(transition))
```

2.1.2 Estimation

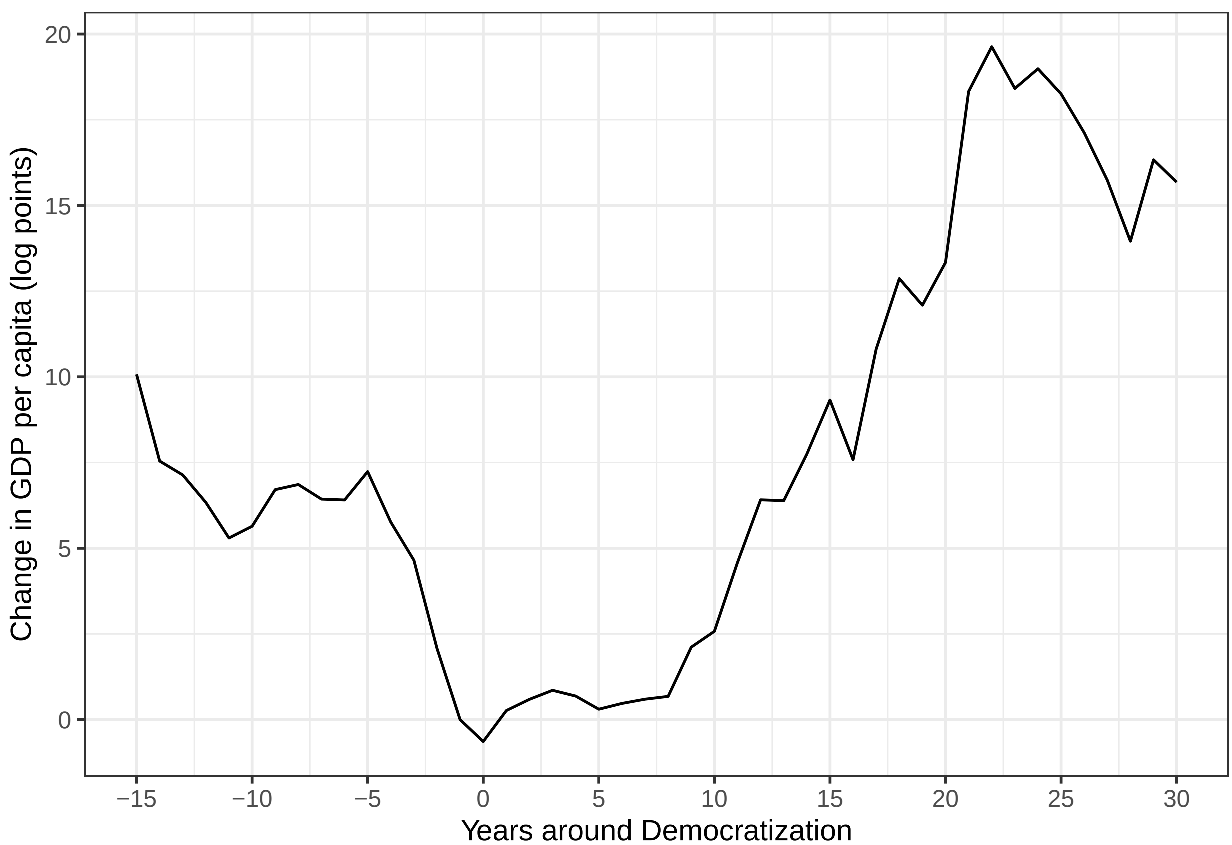
```r
# Define a function to estimate the Average Treatment Effect on the Treated
# (ATET)
estimateATET <- function(outcome_col) {
    # Filter out rows with missing outcome or transition values
    sub_data <- data_f1 %>%
        filter(!is.na(.data[[outcome_col]]), !is.na(transition))
    if (nrow(sub_data) == 0)
```

```r
        return(NA)

    # Create a factor for 'year' with sorted levels
    year_levels <- sort(unique(sub_data$year))
    sub_data <- sub_data %>%
        mutate(year_factor = factor(year, levels = year_levels))

    # Split the data into control (transition == 0) and treated (transition ==
    # 1) groups
    control_data <- sub_data %>%
        filter(transition == 0)
    treated_data <- sub_data %>%
        filter(transition == 1)

    # Return NA if the control group lacks sufficient observations or year
    # variation
    if (nrow(control_data) < 2 || length(unique(control_data$year)) < 2)
        return(NA)

    # Fit a linear model on the control group with year dummies (without
    # intercept)
    model_formula <- as.formula(paste(outcome_col, "~ year_factor - 1"))
    control_model <- tryCatch(lm(model_formula, data = control_data), error = function(e) NULL)
    if (is.null(control_model))
        return(NA)

    # Predict outcomes for the treated group using the control model
    predicted_outcomes <- tryCatch(predict(control_model, newdata = treated_data),
        error = function(e) rep(NA, nrow(treated_data)))

    # Compute treatment effects as the difference between actual and predicted
    # outcomes
    treatment_effects <- treated_data[[outcome_col]] - predicted_outcomes

    # Return the mean treatment effect on the treated (ATET)
    mean(treatment_effects, na.rm = TRUE)
}

# Define relative time periods: pre-treatment (-15 to -1) and post-treatment (0
# to 30)
relative_times <- c(seq(-15, -1), seq(0, 30))
atets <- numeric(length(relative_times))

# Loop over each relative time period to estimate ATET for the corresponding
# outcome column
for (i in seq_along(relative_times)) {
    t_val <- relative_times[i]
    # Set the column name based on whether the period is before (m), during
    # (0), or after (p) treatment
    if (t_val < 0) {
        col_name <- paste0("gdpDiff_m", abs(t_val))
    } else {
        col_name <- if (t_val == 0)
```

```r
            "gdpDiff_0" else paste0("gdpDiff_p", t_val)
    }
    atets[i] <- estimateATET(col_name)
}

# Create a data frame with the relative time periods and their corresponding
# ATET estimates
results_df <- data.frame(RelativeTime = relative_times, ATET = atets)
```

2.1.3 Plot

```r
ggplot(results_df, aes(x = RelativeTime, y = ATET)) + geom_line(color = "black") +
    scale_x_continuous(breaks = seq(-15, 30, 5)) + labs(x = "Years around Democratization",
    y = "Change in GDP per capita (log points)") + theme_bw()
```



2.2 Table.1

2.3 Table.2

2.4 Figure.2

2.5 Table.3

2.6 Table.4

2.7 Table.5

3 Extention