GDP Deflator Development 2000-2022

Daya Ha-Thanh-Van Dau

The graph is used for the M-SD 1 7102: Development Economics at Hochschule Rhein-Waal. Therefore the project cannot be publicly on GitHub. If you want to re-run the code, please first download the Excel file: **historical_classification_by_income.xlsx** from the **repository**.

It aims to observe the differences in GDP deflation development grouped by income level in 2000. That is, freeze the development of the countries and observe only the GDP deflation development.

More projects at **GitHub Profile**

Detailed Portfolio on **Notion Website**

Code with explanation

Load libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(WDI)
library(tidyr)
library(ggplot2)
library(readxl)
library(cellranger)
library(simputation)
library(imputeTS)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.3.3
```

```
library(pandoc)
```

Prepare income class 2000 classification

The classification data is taken out from Excel sheet provided by World Bank DataBank.

```
Inc2000 <- read_excel("historical_classification_by_income.xlsx", "Country Analytical History")
```

```
## New names:
## * `` -> `...1`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...11`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...14`
## * `` -> `...15`
## * `` -> `...16`
## * `` -> `...17`
## * `` -> `...18`
## * `` -> `...19`
## * `` -> `...20`
## * `` -> `...21`
## * `` -> `...22`
## * `` -> `...23`
## * `` -> `...24`
## * `` -> `...25`
## * `` -> `...26`
## * `` -> `...27`
## * `` -> `...28`
## * `` -> `...29`
## * `` -> `...30`
## * `` -> `...31`
## * `` -> `...32`
## * `` -> `...33`
## * `` -> `...34`
## * `` -> `...35`
## * `` -> `...36`
## * `` -> `...37`
```

```r
income2000 <- Inc2000[c(1,11:238),c(1,2,16)]

income2000 <- income2000 %>%
  rename(incomeold = ...16) %>%
  rename(iso3c = ...1) %>%
  filter(incomeold!="..")

income2000$incomeold <- factor(
  income2000$incomeold,
  levels = c("L", "LM", "UM", "H"),
  labels = c("Low income (2000)",
             "Lower middle income (2000)",
             "Upper middle income (2000)",
             "High income (2000)"))
```

Load data

Access to WDI databank for most updated data.

```r
if(!exists("WDI_df")) {
  WDI_df <- WDI(indicator = c("NY.GDP.DEFL.KD.ZG"),
                start = 2000,
                end = 2022,
                extra = TRUE)
}
```

Rename and only select needed attributes.

```r
gdpdfl_2000 <- WDI_df %>%
  rename(GDPdefl = NY.GDP.DEFL.KD.ZG) %>%
  select(country, iso3c, year, GDPdefl, income) %>%
  subset(income != "Aggregates" & !(iso3c == "COD" & year == "2000" ))
```

Data Processing

First will observe the missing value status of the dataset. Then decided how to filter out data then use linear regression to impute.

Filter out NA

```r
# income2000 is the data with the old income category
merge_gdpdefl <- merge(gdpdfl_2000, income2000, by="iso3c", all.x=FALSE)

# Check for missing values
statsNA(merge_gdpdefl$GDPdefl)
```

```
## [1] "Length of time series:"
## [1] 4645
## [1] "-------------------------"
```

```
## [1] "Number of Missing Values:"
## [1] 143
## [1] "-------------------------"
## [1] "Percentage of Missing Values:"
## [1] "3.08%"
## [1] "-------------------------"
## [1] "Number of Gaps:"
## [1] 55
## [1] "-------------------------"
## [1] "Average Gap Size:"
## [1] 2.6
## [1] "-------------------------"
## [1] "Stats for Bins"
## [1] "  Bin 1 (1162 values from 1 to 1162) :      42 NAs (3.61%)"
## [1] "  Bin 2 (1162 values from 1163 to 2324) :      25 NAs (2.15%)"
## [1] "  Bin 3 (1162 values from 2325 to 3486) :      35 NAs (3.01%)"
## [1] "  Bin 4 (1159 values from 3487 to 4645) :      41 NAs (3.54%)"
## [1] "-------------------------"
## [1] "Longest NA gap (series of consecutive NAs)"
## [1] "23 in a row"
## [1] "-------------------------"
## [1] "Most frequent gap size (series of consecutive NA series)"
## [1] "1 NA in a row (occurring 25 times)"
## [1] "-------------------------"
## [1] "Gap size accounting for most NAs"
## [1] "3 NA in a row (occurring 10 times, making up for overall 30 NAs)"
## [1] "-------------------------"
## [1] "Overview NA series"
## [1] "  1 NA in a row: 25 times"
## [1] "  2 NA in a row: 14 times"
## [1] "  3 NA in a row: 10 times"
## [1] "  4 NA in a row: 1 times"
## [1] "  6 NA in a row: 1 times"
## [1] "  7 NA in a row: 1 times"
## [1] "  8 NA in a row: 1 times"
## [1] "  12 NA in a row: 1 times"
## [1] "  23 NA in a row: 1 times"
```

```r
# Create group of countries only have 0 or 1 observations
many_na_countries_2000 <- merge_gdpdefl %>%
  filter(is.na(GDPdefl)) %>%
  group_by(country) %>%
  summarise(n()) %>%
  filter(`n()` >= 23)

# Filter out them from the main dataset
many_na_countries_list_2000 <- many_na_countries_2000$country

dataGDPdeflator_2000 <- merge_gdpdefl %>%
  mutate(drop = ifelse(country %in% many_na_countries_list_2000, T, F)) %>%
  filter(drop == F) %>%
  select(-drop)

statsNA(dataGDPdeflator_2000$GDPdefl)
```

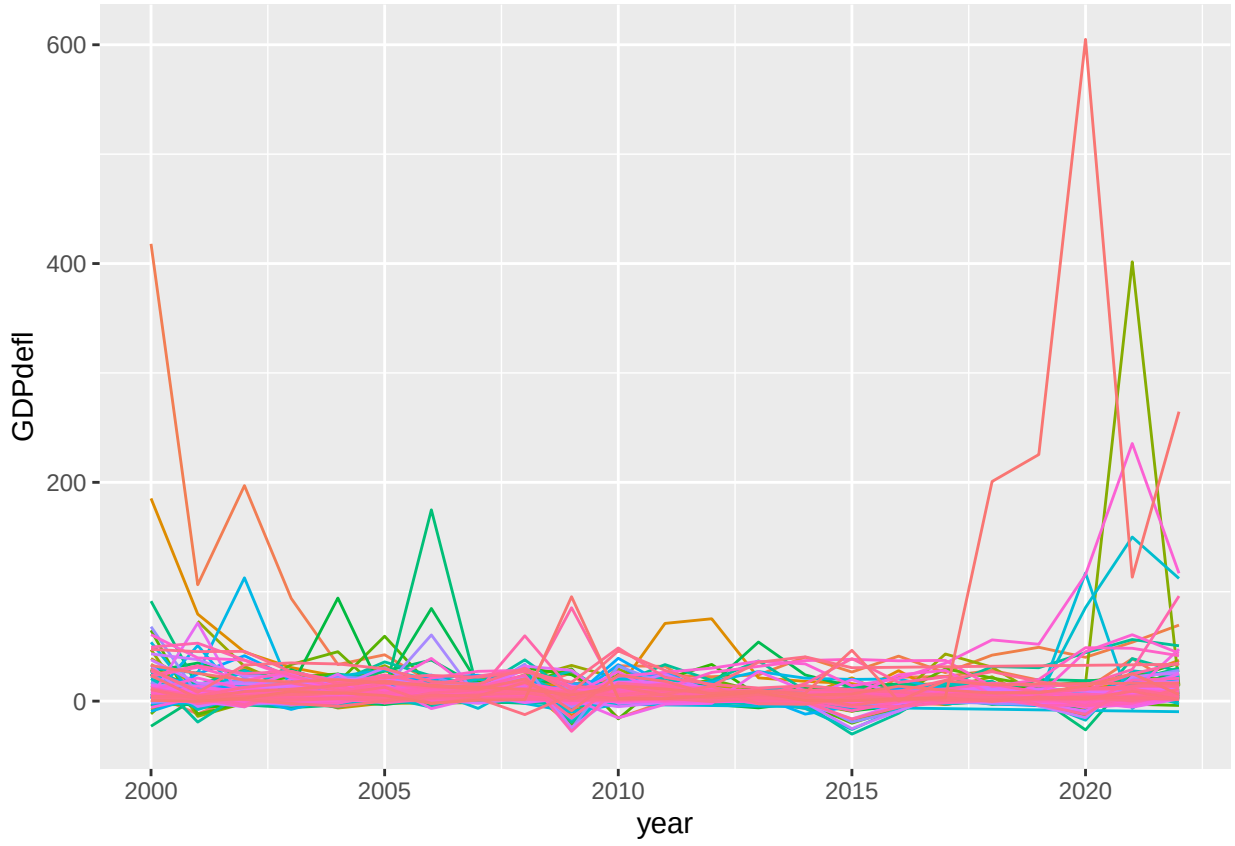
```
## [1] "Length of time series:"
## [1] 4622
## [1] "-------------------------"
## [1] "Number of Missing Values:"
## [1] 120
## [1] "-------------------------"
## [1] "Percentage of Missing Values:"
## [1] "2.6%"
## [1] "-------------------------"
## [1] "Number of Gaps:"
## [1] 54
## [1] "-------------------------"
## [1] "Average Gap Size:"
## [1] 2.222222
## [1] "-------------------------"
## [1] "Stats for Bins"
## [1] "  Bin 1 (1156 values from 1 to 1156) :      42 NAs (3.63%)"
## [1] "  Bin 2 (1156 values from 1157 to 2312) :      25 NAs (2.16%)"
## [1] "  Bin 3 (1156 values from 2313 to 3468) :      35 NAs (3.03%)"
## [1] "  Bin 4 (1154 values from 3469 to 4622) :      18 NAs (1.56%)"
## [1] "-------------------------"
## [1] "Longest NA gap (series of consecutive NAs)"
## [1] "12 in a row"
## [1] "-------------------------"
## [1] "Most frequent gap size (series of consecutive NA series)"
## [1] "1 NA in a row (occurring 25 times)"
## [1] "-------------------------"
## [1] "Gap size accounting for most NAs"
## [1] "3 NA in a row (occurring 10 times, making up for overall 30 NAs)"
## [1] "-------------------------"
## [1] "Overview NA series"
## [1] "  1 NA in a row: 25 times"
## [1] "  2 NA in a row: 14 times"
## [1] "  3 NA in a row: 10 times"
## [1] "  4 NA in a row: 1 times"
## [1] "  6 NA in a row: 1 times"
## [1] "  7 NA in a row: 1 times"
## [1] "  8 NA in a row: 1 times"
## [1] "  12 NA in a row: 1 times"
```

Imputation for NA

```
# Linear Regression approach

simpdataGDPdeflator_2000 <- impute_lm(dataGDPdeflator_2000, GDPdefl ~ year*country)

## After imputation
ggplot(simpdataGDPdeflator_2000, aes(x = year, y = GDPdefl, color = country)) +
   geom_line(stat = "identity", show.legend = F)
```




```
summary(simpdataGDPdeflator_2000)
```

```
##     iso3c              country                year         GDPdefl
##  Length:4622        Length:4622        Min.   :2000   Min.   :-30.200
##  Class :character   Class :character   1st Qu.:2005   1st Qu.:  1.472
##  Mode  :character   Mode  :character   Median :2011   Median :  3.721
##                                        Mean   :2011   Mean   :  6.910
##                                        3rd Qu.:2017   3rd Qu.:  8.004
##                                        Max.   :2022   Max.   :604.946
##     income           World Bank Analytical Classifications
##  Length:4622        Length:4622
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##                        incomeold
##  Low income (2000)          :1402
##  Lower middle income (2000):1219
##  Upper middle income (2000): 828
##  High income (2000)         :1173
##
##
```

Visualisation

After process the dataset, first will transform by building mean then visualisa in the line graph.

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```