Advanced Macroeconometrics – Assignment 1

Siegfried Hammer (h12229325@wu.ac.at) Max Heinze (h11742049@wu.ac.at)
Tim Koenders (h12215486@wu.ac.at)

April 19, 2023

The executable code that was used in compiling the assignment is available on GitHub at https://github.com/maxmheinze/macrometrics.

Exercise 1

First, we read in the data set, removing the first column indicating the recommended transformation.

```
# Header ------------------------------------------------------------------

rm(list = ls())
gc()

pacman::p_load(tidyverse, urca)


# Read in Data ------------------------------------------------------------

fred <- read.csv("./assignment1/data/fred.csv")[-1, ]
```

Next, we create the desired function `ts_explode()`. Along with the vector to be transformed, it asks for a specification of whether the data is ordered with the latest or earliest value first. It takes the earliest value first as default.

```
# Create the function -----------------------------------------------------

ts_explode <- function(input_vector, start_with_latest = FALSE) {
    # The function is called ts_explode because a single vector explodes into
    # an entire data frame. Boom!

    # Package dplyr required for lag() function
    require(dplyr)

    # Reverse input vector if user specifies it is sorted latest to earliest
    input_vector <- if (start_with_latest == FALSE) {
        input_vector
    } else {
        rev(input_vector)
    }

    # Do the transformations, assign transformed vectors
    original <- input_vector
    log_transformed <- log(input_vector)
    mom_growth <- input_vector/dplyr::lag(input_vector, 1) - 1
    yoy_growth <- input_vector/dplyr::lag(input_vector, 12) - 1
    yoy_growth_lagged <- dplyr::lag(input_vector, 12)/dplyr::lag(input_vector, 24) -
        1
```

```
    # Create a data frame to export, reverse ordering back to original in case
    # start_with_latest = TRUE was specified
    export_df <- if (start_with_latest == FALSE) {
        data.frame(original, log_transformed, mom_growth, yoy_growth, yoy_growth_lagged)
    } else {
        data.frame(original = rev(original), log_transformed = rev(log_transformed),
            mom_growth = rev(mom_growth), yoy_growth = rev(yoy_growth), yoy_growth_lagged =
            rev(yoy_growth_lagged))
    }

    # Display warnings regarding ordering and units of growth rates
    warning("By default, ts_explode() assumes that values are ordered from earliest to
    latest. If your vector is ordered from latest to earliest, specify `start_with_latest =
    TRUE`!")
    warning("Growth rates are given in decimals, not in percent!")

    # Return the data frame
    return(export_df)
}
```

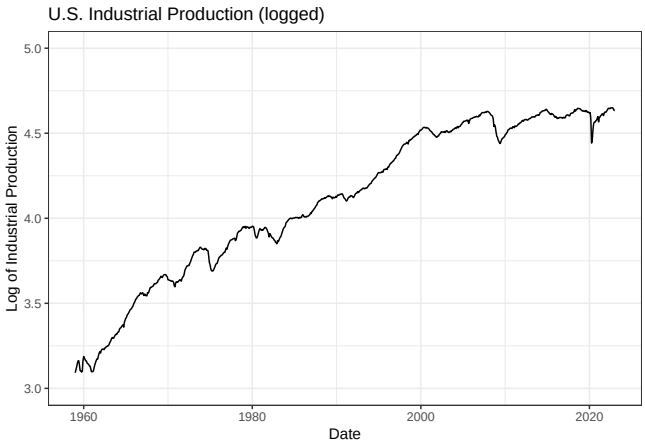
Using `ts_explode()`, we create the data frame `ind_prod` including all transformations of the `INDPRO` variable. We bind the data frame together with the date column, which we transform from character to date. All other changes in the resulting data frame are of cosmetic nature.

```
# Prepare Industrial Production Data Frame ---------------------------------

ind_prod <- fred$sasdate %>%
    cbind(ts_explode(fred$INDPRO)) %>%
    as_tibble() %>%
    mutate(date = lubridate::mdy(.)) %>%
    select(-.) %>%
    relocate(date, .before = original)
```

Next, we plot both the logged variable and the year-on-year growth rate.

```
# Create Log Plot ----------------------------------------------------------

ind_prod %>%
    ggplot() + geom_line(aes(x = date, y = log_transformed)) + labs(title = "U.S. Industrial
    Production (logged)",
    x = "Date", y = "Log of Industrial Production") + ylim(3, 5) + theme_bw()
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```




```
# Create Growth Plot -------------------------------------------------------

ind_prod %>%
```

```
    ggplot() + geom_line(aes(x = date, y = yoy_growth)) + labs(title = "U.S. Industrial
    Production (year-on-year growth)",
    x = "Date", y = "Year-on-Year Growth of Industrial Production") + theme_bw()
```

```
## Warning: Removed 13 rows containing missing values (`geom_line()`).
```