Economic Data 2023 - Labs

Week 8

1 Part 1

In this exercise we will plot the unemployment rate for the UK since 1890. To solve the exercise, go through the following steps.

- Create a new R project and name it `lab_7`. This is done by navigating as follows `file/New Project/New Directory/New Project`. In the `lab_7` folder, create new folders to organize it in the same way as the `econ_data_project_template` from last week.
- Download the dataset `AURUKM.csv` from Blackboard and import the dataset into R.
- Create a plot of the evolution of the unemployment rate of the UK since 1890.
- What year had the highest unemployment rate in the UK since 1890?

2 Part 2

In this exercise you will decompose the unemployment rate of your assigned country. We will use data from the OECD to construct the below charts. For this exercise, the solutions to last week as well as the lecture slides from this week might be useful. As always, your country depends on the timing of your lab session:

- Group 1: Tuesday 10:00 to 12:00: Slovenia
- Group 2: Wednesday 9:00 to 11:00: Greece
- Group 4: Wednesday 11:00 to 13:00: Finland
- Group 3: Thursday 9:00 to 11:00: Spain

To solve the exercise, go through the following steps.

- Go to OECD and download data on the unemployment rate for your country. In the database, this can be found under labor force statistics/short-term labor force statistics/Monthly unemployment rates. Download data for all persons, seasonally unadjusted, monthly, since January 2014. Download the data as a .csv file and save it in the folder `lab_7/data`.

- Create a script named `data_prep.R` and load the dataset into R.
- Clean the data by selecting the relevant columns and renaming the columns to more informative names. Name the column denoting the time `date`.
- Note that in the column `date`, the day of the month is missing. The date the data is recorded is the first of each month. Change the column to add this information. Hint: `mutate(date = paste(date, "01", sep="-"))`.
- To specify that we are working with dates, we need to encode the column `date` as a date. In the data, we see that the date format is year-month-day. To specify this you can use mutate and `date = as.Date(date, format = "%Y-%m-%d")`.
- Create a new column in your dataset called trend using a two-period moving average. The two-period moving average is given by the following formula

$$\text{trend}_t = \frac{1}{5} \sum_{i=t-2}^{t+2} u_i, \tag{1}$$

  where u_i denotes the unemployment at time i.
- Create a new column in your dataset which is the difference between the unemployment rate and the trend.
- Keep only rows that are recorded after January 1. 2014. Hint: `date>'2014-01-01'`.
- Reproduce Figure 1. To combine the three plots, first generate the plots separaetly and assign them the names `p1`, `p2`, `p3`. Then import the library `ggpubr` and use the command `ggarrange(p1, p2, p3 ,ncol = 1, nrow = 3)`.
- Download the same unemployment data as above but now *both* seasonally adjusted and unadjusted.
- Load, clean, and prepare the new dataset in the same manner as above.
- Create a new column in your dataset called `adj` which equals `No seasonal adjustment` if the row is seasonally adjusted and `Seasonal adjustment` otherwise. Hint: Use the `mutate()` and `ifelse()` functions.
- Reproduce Figure 2.

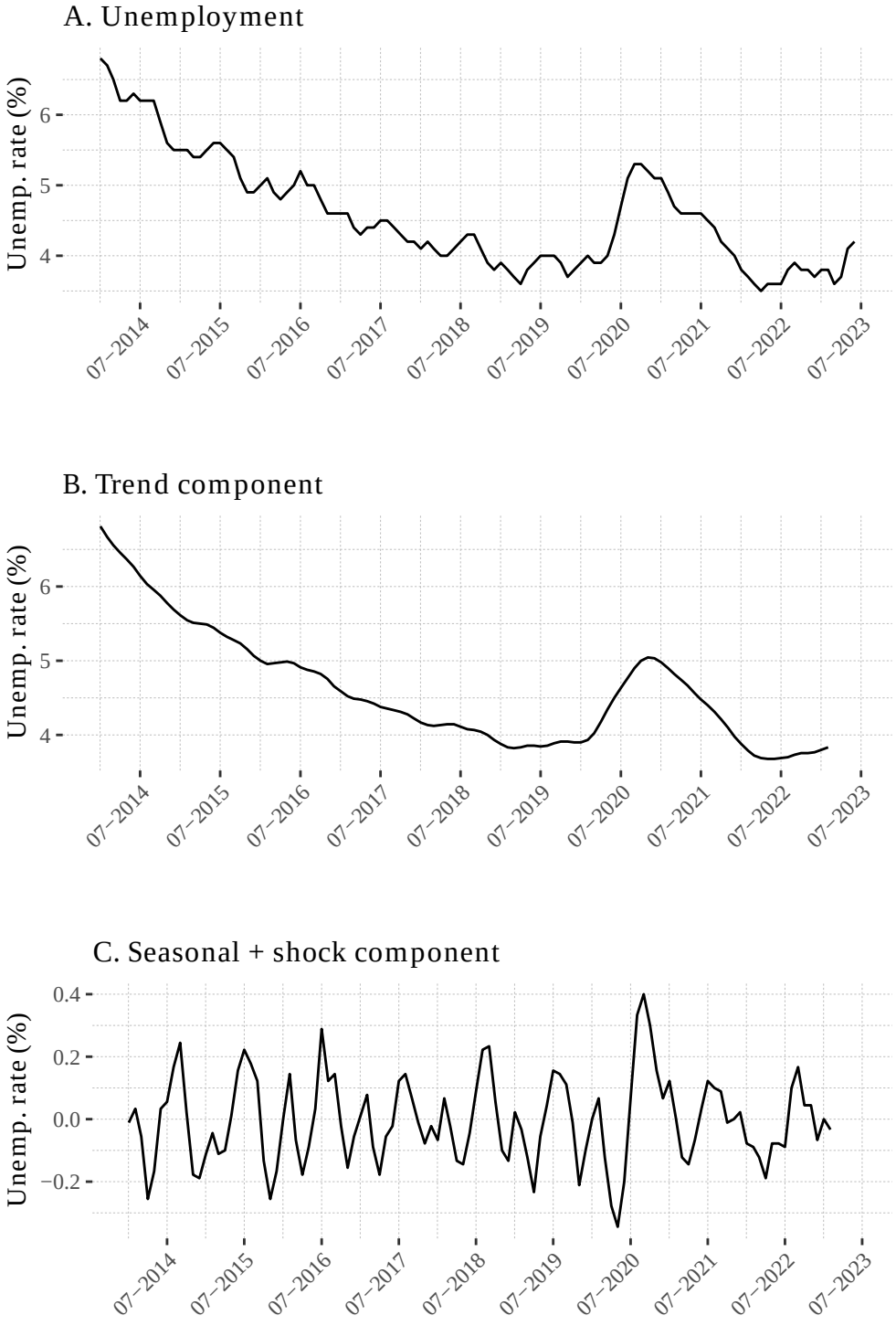

Figure 1: Decomposition of unemployment for the UK.

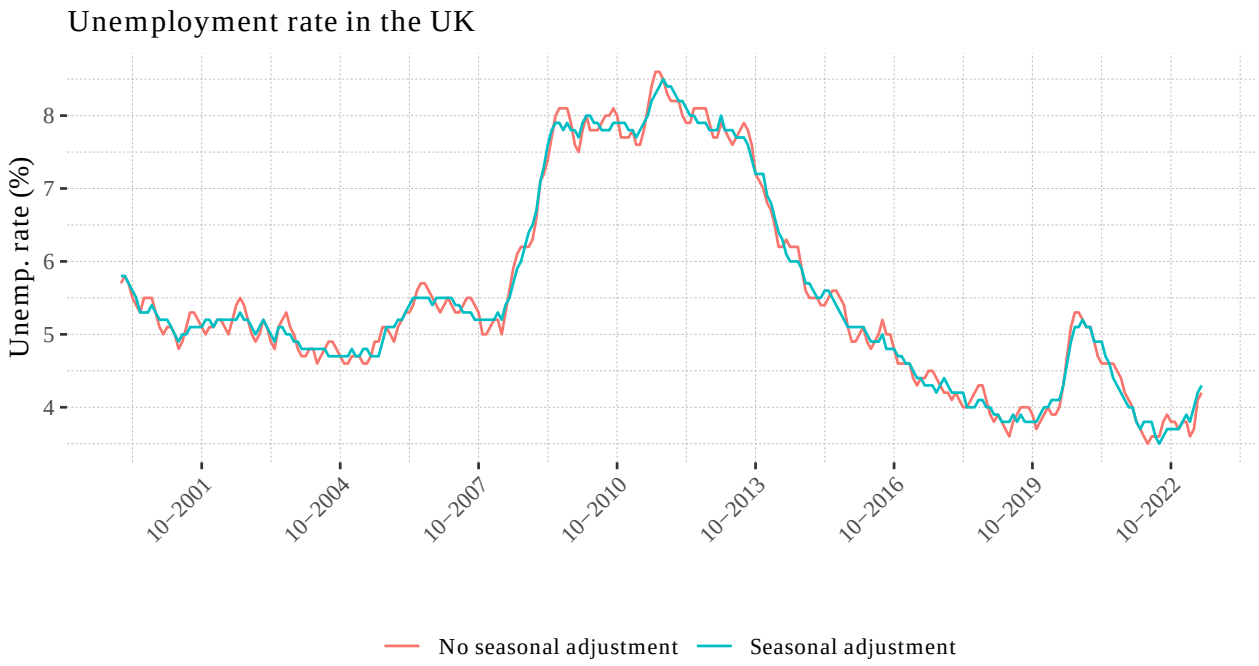

Figure 2: Seasonal decomposition of unemployment for the UK.