Neural additive VAR

Proposal

Marc Agustí (marc.agusti@barcelonagse.eu)
Patrick Altmeyer (patrick.altmeyer@barcelonagse.eu)
Ignacio Vidal-Quadras Costa (ignacio.vidalquadrascosta@barcelonagse.eu)

May, 2021

1 Loading and merging data

Below I just load and merge the data for the US given the .csv files you moved in the `data_VAR` folder.

> NOTE: Eventually we want to merge date for other countries in here as well to have one clean data frame in long (tidy) format to work with. — Pat

```
library(data.table)
data_files <- list.files("data_VAR")
countries <- c("US") # add more as more data available
dt <- rbindlist(
  lapply(
    countries,
    function(country) {
      data_files_country <- data_files[grepl(country,data_files,ignore.case = TRUE)] # country-level da
      rbindlist(
        lapply(
          data_files_country,
          function(file_path) {
            dt <- fread(file.path("data_VAR", file_path))
            setnames(dt, colnames(dt), c("date", "value"))
            variable <- file_path
            for (pattern in c("USA", "US",".csv","_")) {
              variable <- stringr::str_remove(variable, pattern) # remove all patterns except variable
            }
            dt[,variable:=variable]
            dt[,country:=toupper(country)]
            return(dt)
          }
        )
      )
    }
  )
)
```

2 Exploring

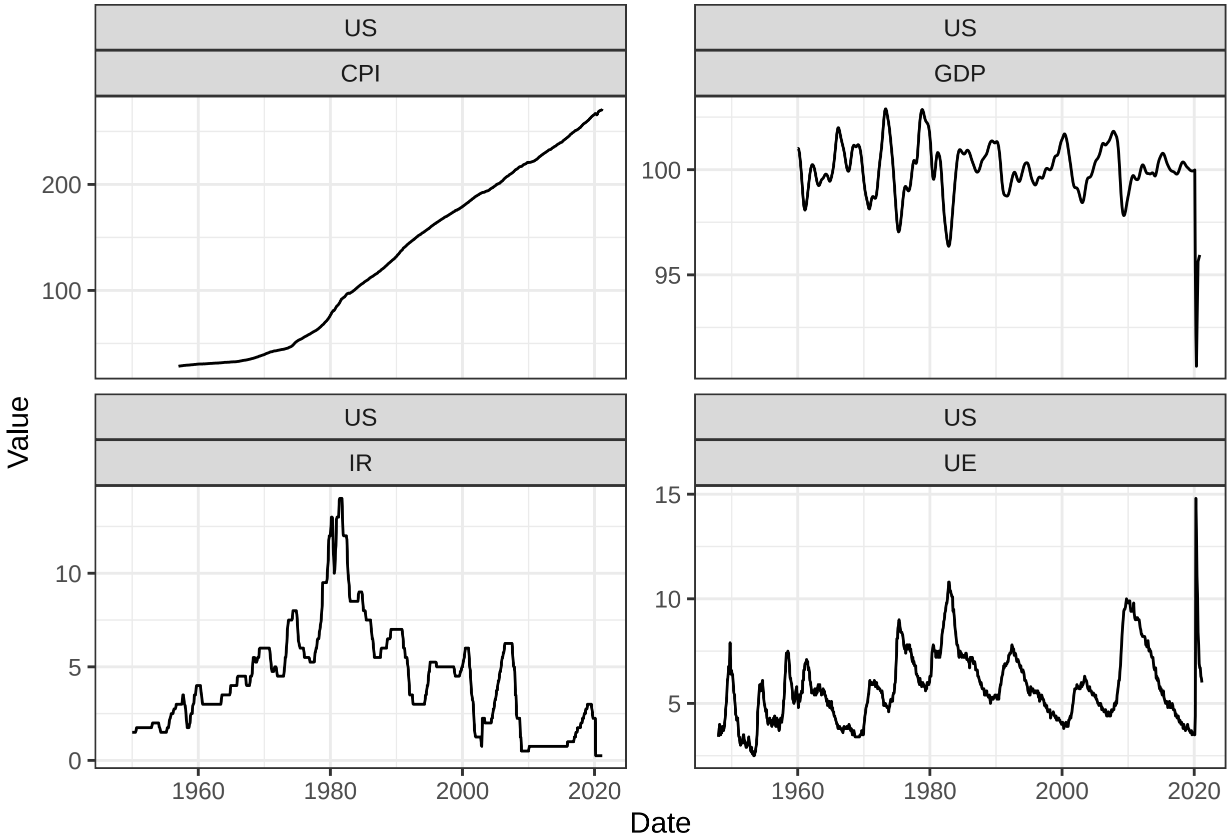
Now let's have a quick look at the data. First of all we note that the date range is slightly different, but that is not a reason for concern.

```
dt[,.(date_range=range(date)),by=.(variable,country)]
```

```
##    variable country date_range
## 1:       IR      US 1950-01-01
## 2:       IR      US 2021-03-01
## 3:       UE      US 1948-01-01
## 4:       UE      US 2021-03-01
## 5:      CPI      US 1957-01-01
## 6:      CPI      US 2021-03-01
## 7:      GDP      US 1960-01-01
## 8:      GDP      US 2020-11-01
```

Let's also quickly inspect the time series visually. Unsurprisingly, lots of things we want to take into account here: non-stationarity, business cycles and trends. `CPI` we probably want to convert into inflation before it enters the VAR.

```
library(ggplot2)
ggplot2::ggplot(dt) +
  ggplot2::geom_line(ggplot2::aes(y=value, x=date)) +
  ggplot2::facet_wrap(country ~ variable, scales = "free_y") +
  ggplot2::theme_bw() +
  ggplot2::labs(
    x="Date",
    y="Value"
  )
```



Let's save a copy of that raw data to disk should we ever want to transform in different ways for different approaches.

```
fwrite(dt, "data_VAR/merged_raw.csv")
```

3 Transforming

3.1 VAR

```
dt <- fread("data_VAR/merged_raw.csv") # load the raw merged data
```

Can formalize this through ADF tests, conintegration (VECM?), ... in case supervisor wants to see that, but for the simple VAR we may just get rid of the obvious non-stationarity.

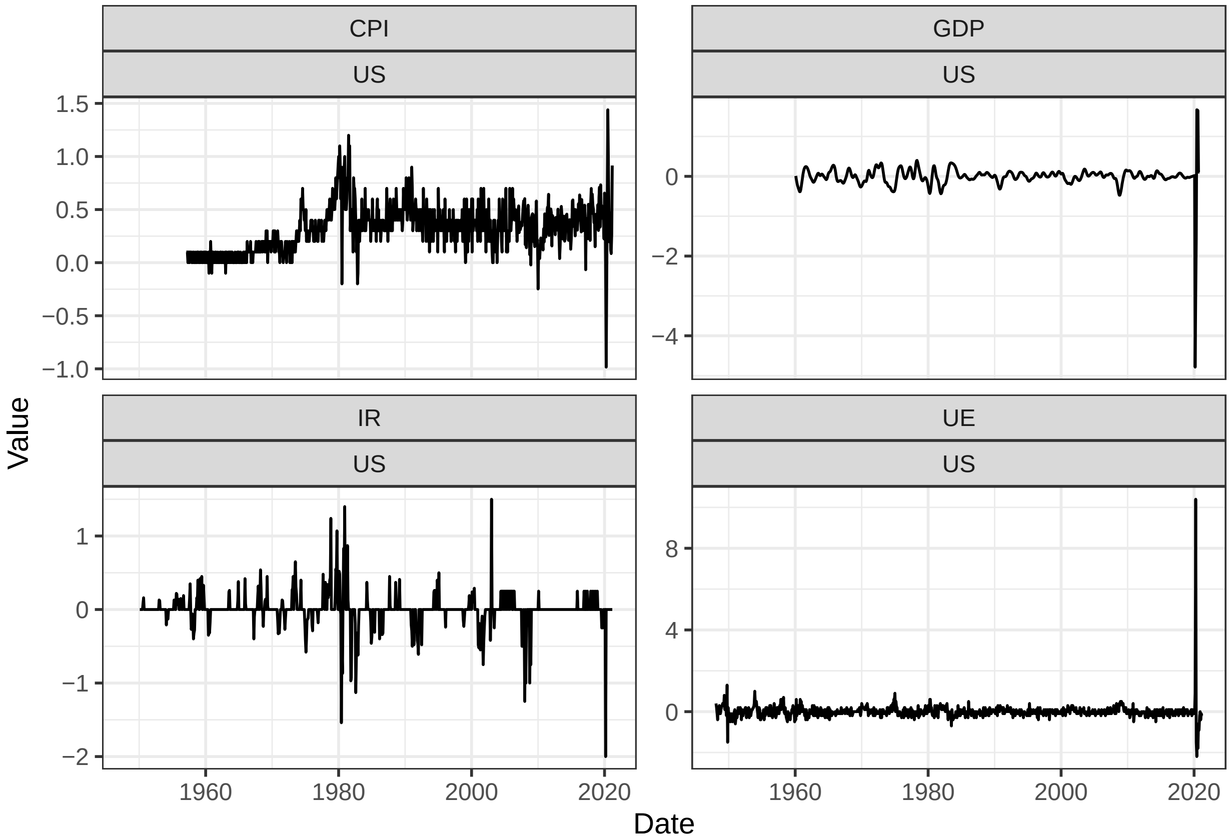
```
dt[,value:=c(NA,diff(value)),by=.(variable,country)]
dt <- na.omit(dt)
```

Looking at this data again, things already look much better in first differences. The `CPI` still exerts some trending behaviour and may need some work. Structural break due to COVID needs to be taking into account in final analysis.

> Consideration: for the NAVAR, should the data be standardizes? — Pat

```
library(ggplot2)
ggplot2::ggplot(dt) +
  ggplot2::geom_line(ggplot2::aes(y=value, x=date)) +
  ggplot2::facet_wrap(variable ~ country, scales = "free_y") +
```

```
  ggplot2::theme_bw() +
  ggplot2::labs(
    x="Date",
    y="Value"
  )
```



Finally, let's turn the data into a wide format to be fed to then later estimate the VAR. Let's also make sure we cover the exact same time frame for all series. We do so by first completing the time data frame with respect to the `date` variable. Then we cast the data across `variable`.

```
library(tidyr)
dt <- data.table(tidyr::complete(dt, date, nesting(variable, country))) # complete data frame wrt date
dt <- dcast(dt, date + country ~ variable, value.var = "value") # cast data
dt
```

```
##            date country   CPI        GDP IR   UE
##   1: 1948-02-01      US    NA         NA NA  0.4
##   2: 1948-03-01      US    NA         NA NA  0.2
##   3: 1948-04-01      US    NA         NA NA -0.1
##   4: 1948-05-01      US    NA         NA NA -0.4
##   5: 1948-06-01      US    NA         NA NA  0.1
##  ---
## 874: 2020-11-01      US 0.469 0.09966331  0 -0.2
## 875: 2020-12-01      US 0.121         NA  0  0.0
## 876: 2021-01-01      US 0.085         NA  0 -0.4
## 877: 2021-02-01      US 0.274         NA  0 -0.1
## 878: 2021-03-01      US 0.915         NA  0 -0.2
```

Now, when we omit NA we will automatically get rid of all rows that contain a missing values for at least one of the time series. Just in case we want to add further visualizations of the preprocessed data we will also save a version of the data in long (tidy) format. Both versions are then saved to disk.

```
dt <- na.omit(dt)
dt_l <- melt(dt, id.vars = c("date", "country"))
fwrite(dt, "data_VAR/preprocessed.csv")
fwrite(dt_l, "data_VAR/preprocessed_tidy.csv")
```