Banknotes in Circulation

Victor Cuspinera

23/10/2020

Banknotes in Circulation

Overview

One of the main activities of Banco de Mexico is to issue new Banknotes and Coins and ensure an adequate level of Banknotes and Coins in Circulation to satisfy the Cash requirements of the people, businesses and companies, to allow the Economy of the country work properly.

The aim of this document is to analyze the Banknotes in Circulation using the SIE API with the `siebanxicor` R-package, and the custome function `sie_function` from the *SIE_function.R* script.

Quick overview

I use the developed customed function for an overview of the first draft to look into the Banknotes in Circulation by denomination.

```
# load libraries
library("siebanxicor")
library("tidyverse")
```

```
## -- Attaching packages ------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ---------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
# call and set the token
token_file <- read.csv("../token/SIE_Token.csv", header=FALSE)
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## incomplete final line found by readTableHeader on '../token/SIE_Token.csv'
```

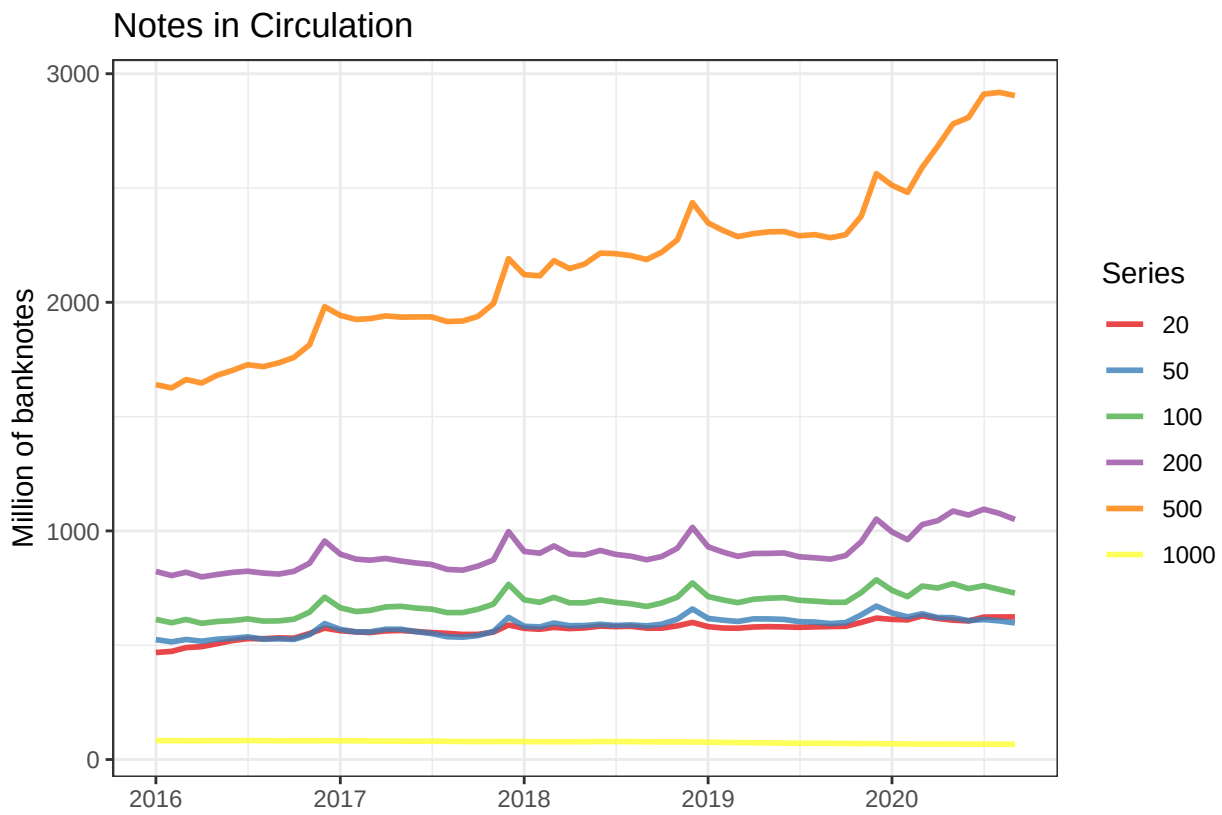
```r
setToken(token_file$V2)

# call the customed function from an RScript
source("SIE_function.R")

# setting the variables for current Notes in Circulation
my_series <- c('SM1472', 'SM1478', 'SM1479', 'SM1480', 'SM1481', 'SM1482')
my_names <- c('20','50','100','200','500','1000')
my_title <- "Notes in Circulation"
my_start <- '2016-01-01'
my_route <- "../img/"

# run the function
series_NIC <- sie_function(my_series, my_names,
                           my_title, route=my_route,
                           y_lab="Million of banknotes",
                           startDate=my_start)
```

```
## Saving 6.5 x 4.5 in image
```



```
##   idSerie                                          title  startDate    endDate
## 1  SM1472    Total of banknotes in circulation 20 pesos 1993-01-01 2020-09-01
## 2  SM1478    Total of banknotes in circulation 50 pesos 1993-01-01 2020-09-01
## 3  SM1481   Total of banknotes in circulation 500 pesos 1993-01-01 2020-09-01
## 4  SM1482 Total of banknotes in circulation 1,000 pesos 1993-01-01 2020-09-01
## 5  SM1480   Total of banknotes in circulation 200 pesos 1993-01-01 2020-09-01
## 6  SM1479   Total of banknotes in circulation 100 pesos 1993-01-01 2020-09-01
##   frequency dataType               unit
## 1   Monthly   Stocks Millions of pieces
## 2   Monthly   Stocks Millions of pieces
## 3   Monthly   Stocks Millions of pieces
## 4   Monthly   Stocks Millions of pieces
## 5   Monthly   Stocks Millions of pieces
## 6   Monthly   Stocks Millions of pieces
```

And this is an example of how does the Notes in Circulation (NIC) tidy data frame looks like:

```
##           date value  serie serie_name
## 337 2020-04-01  67.5 SM1482       1000
## 338 2020-05-01  67.6 SM1482       1000
## 339 2020-06-01  67.3 SM1482       1000
## 340 2020-07-01  67.2 SM1482       1000
## 341 2020-08-01  66.9 SM1482       1000
## 342 2020-09-01  66.5 SM1482       1000
```

Getting a squared data frame

Tidy data is not always the easiest way to look at through the human-eye. The next code chunk wranges the previously obtained tidy data frame into a messy and square data frame with one column per denomination.

```r
# get square data frame
squared_NIC <- dcast(series_NIC, date~serie)

# rename the series
names(squared_NIC) <- c("date", my_names)

# order by date
squared_NIC <- squared_NIC[order(as.Date(squared_NIC$date, format="%Y-%m-%d")),]

# print an example of the data frame
squared_NIC %>% head()
```

```
##         date    20    50   100   200    500 1000
## 1 2016-01-01 468.1 524.5 613.0 822.7 1640.0 82.7
## 2 2016-02-01 473.0 514.6 598.3 804.6 1625.4 82.3
## 3 2016-03-01 489.4 525.0 612.6 819.2 1662.3 82.1
## 4 2016-04-01 493.7 517.5 595.6 798.9 1646.8 82.1
## 5 2016-05-01 506.3 526.1 603.6 809.2 1680.5 82.4
## 6 2016-06-01 520.4 530.2 607.7 818.6 1702.2 82.2
```

Looking into the annual growth

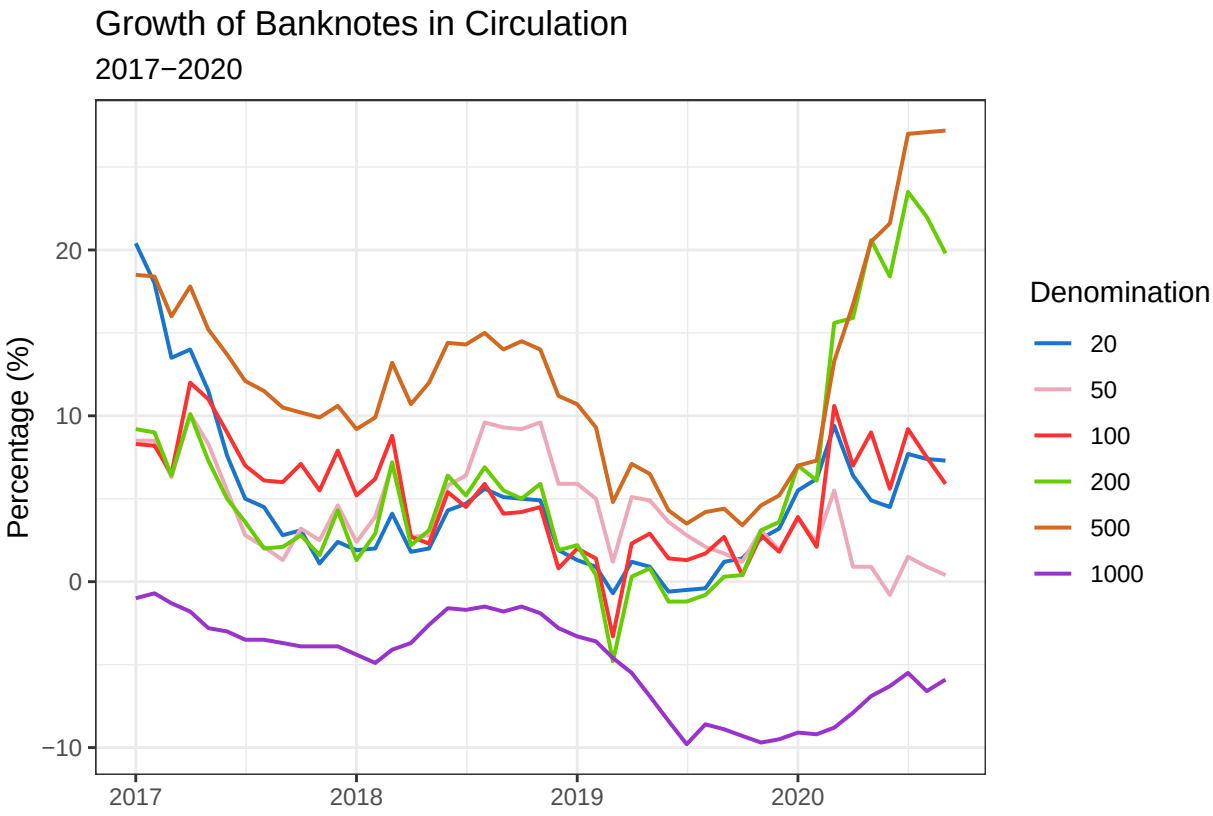
It would be also useful to look into the annual growth (comparing the number of NIC of a specific month vs. the data of the same month from the previous year).

```
# get the growth rate
growth_NIC <- series_NIC %>%
    group_by(serie) %>%
    mutate(annual_growth = round((value - lag(value, 12)) * 100 / lag(value, 12), 1)) %>%
    subset(select = -c(value)) %>%
    drop_na()

# print an example of the data frame
growth_NIC %>% tail()
```

```
## # A tibble: 6 x 4
## # Groups:   serie [1]
##   date       serie  serie_name annual_growth
##   <date>     <chr>  <fct>              <dbl>
## 1 2020-04-01 SM1482 1000                -7.9
## 2 2020-05-01 SM1482 1000                -6.9
## 3 2020-06-01 SM1482 1000                -6.3
## 4 2020-07-01 SM1482 1000                -5.5
## 5 2020-08-01 SM1482 1000                -6.6
## 6 2020-09-01 SM1482 1000                -5.9
```

```
# plot
growth_NIC %>% ggplot() +
  geom_line(aes(x=date, y=annual_growth, color=serie_name), size=0.7) +
  labs(title="Growth of Banknotes in Circulation", subtitle="2017-2020", x="", y="Percentage (%)", colo
  scale_color_manual(values=c("dodgerblue3", "pink2", "firebrick1", "chartreuse3", "chocolate", "darkor
  theme_bw()
```


Growth of Banknotes in Circulation
2017−2020
Percentage (%)
20
10
0
−10
2017
2018
2019
2020
Denomination
20
50
100
200
500
1000