Banknotes in Circulation

Victor Cuspinera

23/10/2020

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 review

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)
```

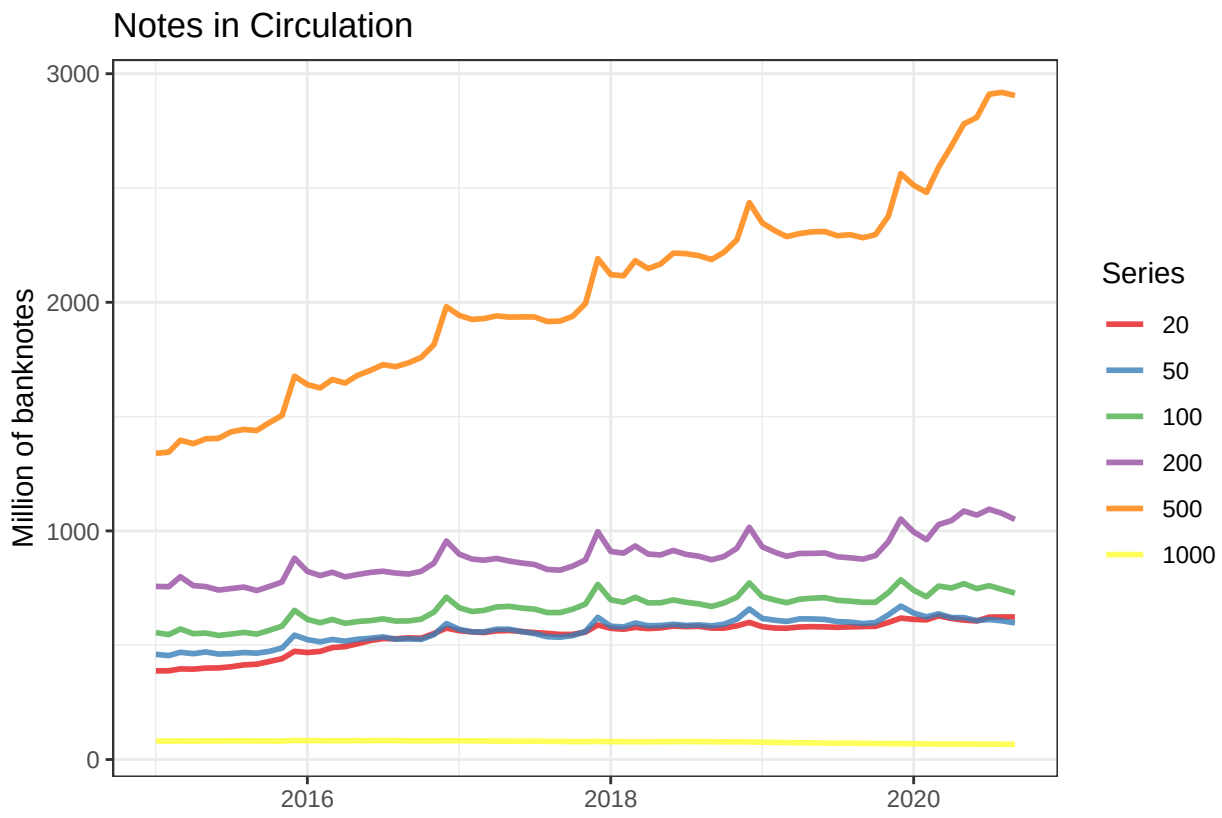
```
##
## Attaching package: 'reshape2'

## 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 <- '2015-01-01'

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  SM1479   Total of banknotes in circulation 100 pesos 1993-01-01 2020-09-01
## 2  SM1482 Total of banknotes in circulation 1,000 pesos 1993-01-01 2020-09-01
## 3  SM1472    Total of banknotes in circulation 20 pesos 1993-01-01 2020-09-01
## 4  SM1478    Total of banknotes in circulation 50 pesos 1993-01-01 2020-09-01
## 5  SM1481   Total of banknotes in circulation 500 pesos 1993-01-01 2020-09-01
## 6  SM1480   Total of banknotes in circulation 200 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
## 409 2020-04-01  67.5 SM1482       1000
## 410 2020-05-01  67.6 SM1482       1000
## 411 2020-06-01  67.3 SM1482       1000
## 412 2020-07-01  67.2 SM1482       1000
## 413 2020-08-01  66.9 SM1482       1000
## 414 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.

```
# 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 2015-01-01 387.7 460.2 555.1 757.2 1339.6 81.0
## 2 2015-02-01 388.0 454.6 546.5 755.8 1344.2 80.8
## 3 2015-03-01 396.2 469.0 571.0 799.7 1396.9 81.0
## 4 2015-04-01 395.3 462.9 550.3 760.6 1381.3 80.8
## 5 2015-05-01 399.9 470.5 553.1 756.6 1402.8 81.3
## 6 2015-06-01 400.5 461.3 542.7 741.1 1404.9 81.2
```

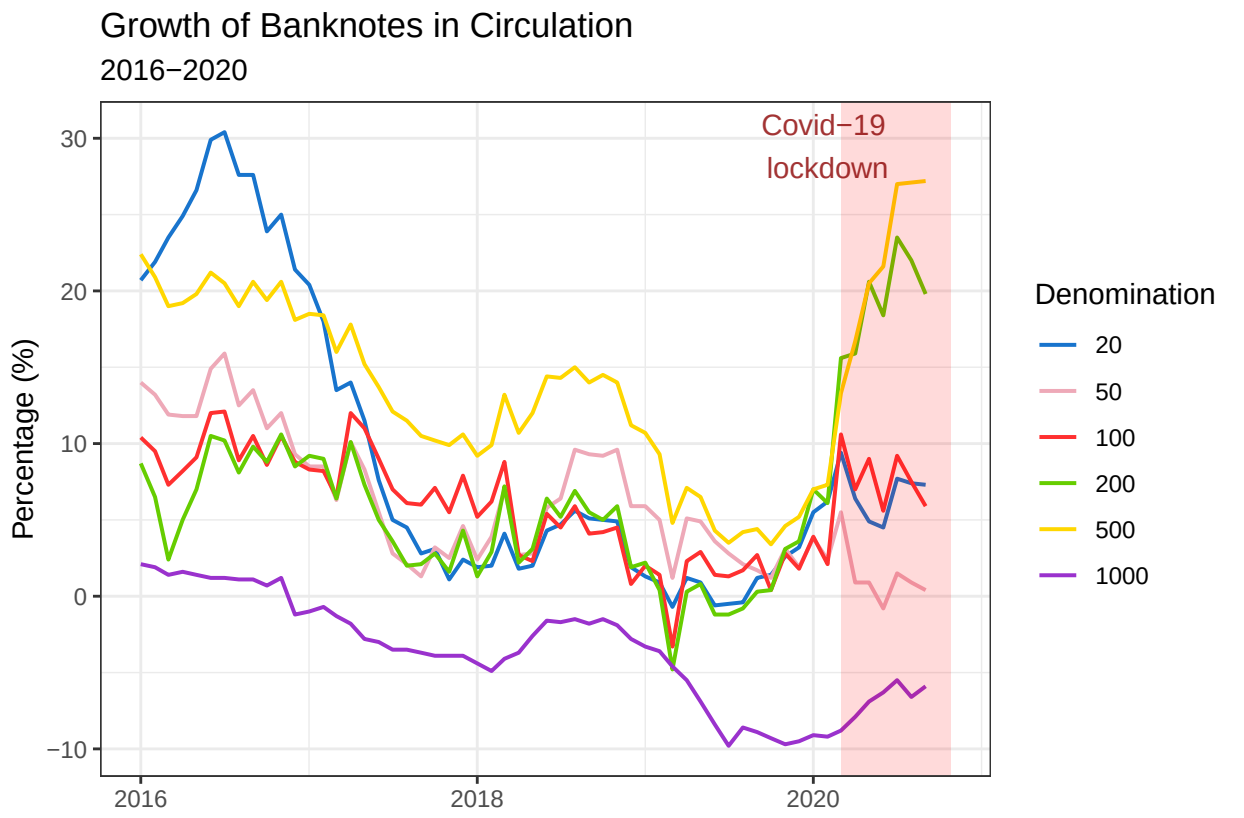
Looking into the annual growth

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 = "2016-2020", x = "", y = "Percentage (%
  scale_color_manual(values = c("dodgerblue3", "pink2", "firebrick1", "chartreuse3", "gold", "darkorchi
    annotate(geom = "rect", xmin = as.Date("2020-03-01", "%Y-%m-%d"), xmax = as.Date(Sys.Date(), "%Y-%m
  annotate("text", x = as.Date("2020-02-01", "%Y-%m-%d"), y = 29.5, label = "Covid-19 \nlockdown", size
  theme_bw()
```




```
# save plot
ggsave(paste0(my_route, "Growth of Banknotes in Circulation.png"))
```

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

Cumulative growth over the year

While the previous graph shows the annual growth of Banknotes in Circulation, a level graph of the cumulative growth of Banknotes in Circulation show us the need of cash during the year.

The first step would be get the Banknotes in Circulation per year.

```
# get the growth rate
df_total_NIC <- series_NIC %>%
  group_by(date) %>%
```

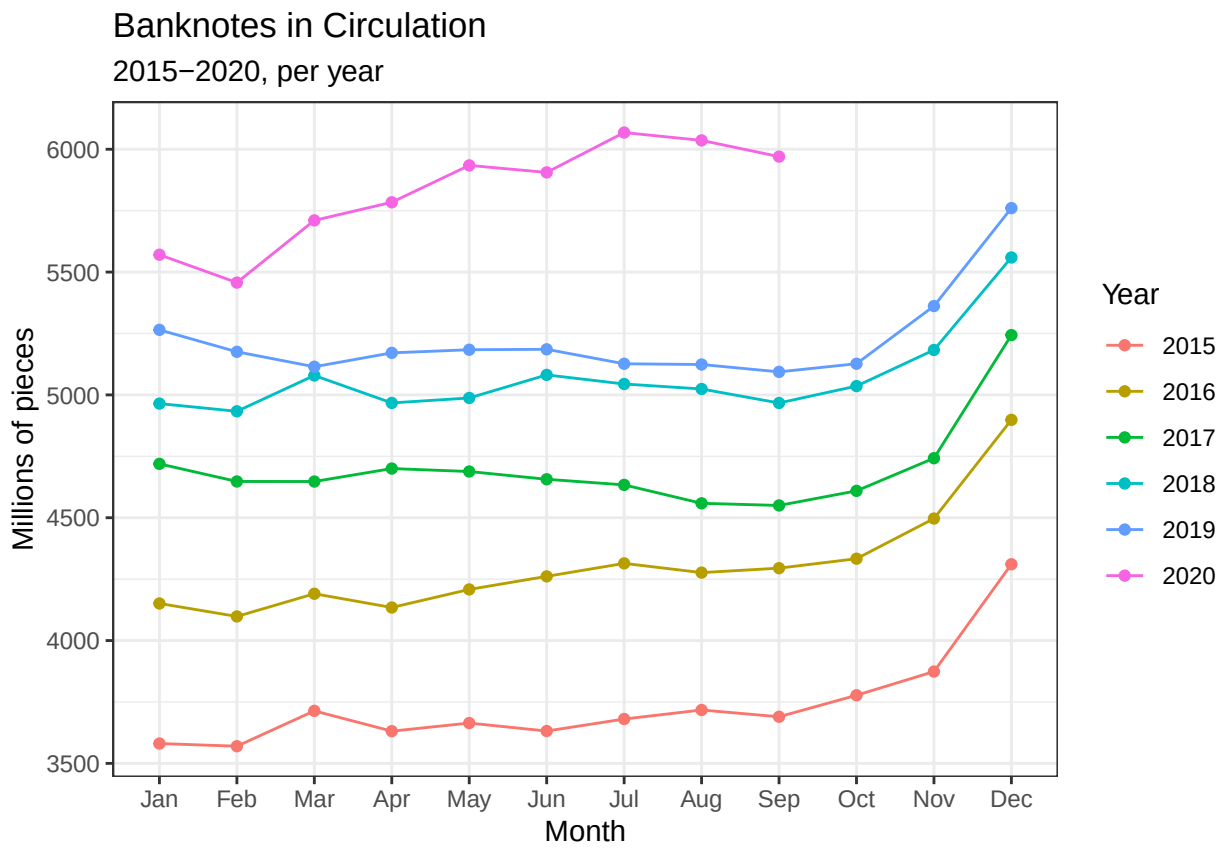
```
  summarise(NIC = sum(value)) %>%
  mutate(year = format(as.Date(date, format = "%Y-%m-%d"),"%Y")) %>%
  mutate(month = format(as.Date(date, format = "%Y-%m-%d"),"%b"))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# set order of months
df_total_NIC$month = factor(df_total_NIC$month, levels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Ju

# show data.frame
df_total_NIC %>% head()
```

```
## # A tibble: 6 x 4
##   date         NIC year  month
##   <date>     <dbl> <chr> <fct>
## 1 2015-01-01 3581. 2015  Jan
## 2 2015-02-01 3570. 2015  Feb
## 3 2015-03-01 3714. 2015  Mar
## 4 2015-04-01 3631. 2015  Apr
## 5 2015-05-01 3664. 2015  May
## 6 2015-06-01 3632. 2015  Jun
```

```
# plot
df_total_NIC %>% ggplot(aes(x = month, y = NIC, color = year, group = year))+
  geom_line() +
  geom_point() +
  labs(title = "Banknotes in Circulation", subtitle = "2015-2020, per year", x = "Month", y = "Millions
  theme_bw()
```



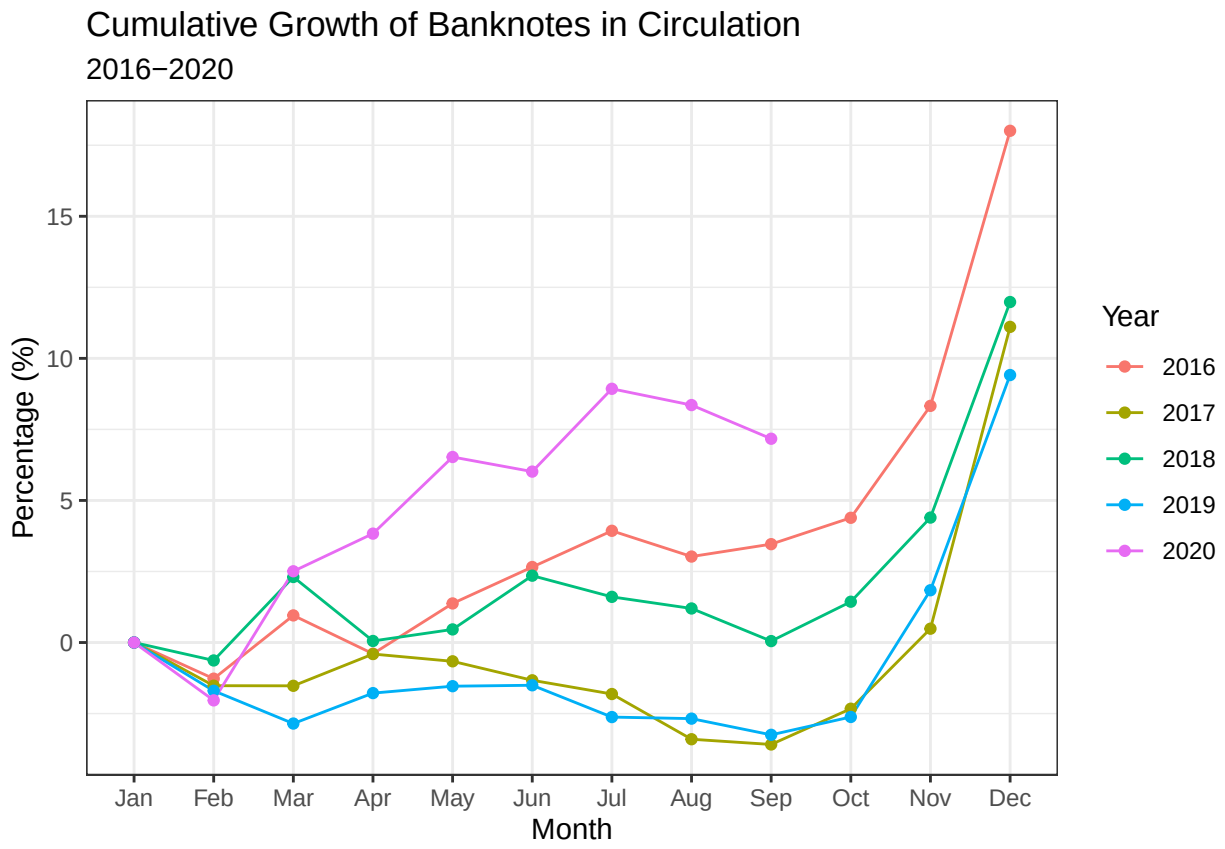

```
# save plot
ggsave(paste0(my_route, "Total Banknotes in Circulation per year.png"))
```

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

The nexyt step is obtain the growth rate using January as starting point.

```
# fix the initial value for each year
df_initial <- df_total_NIC %>%
  #mutate(NIC_diff = NIC - lag(NIC) ) %>%
  melt(id = c("month", "year"), measure="NIC") %>%
  dcast(month ~ year) %>%
  subset(select = c("month", "2016", "2017", "2018", "2019", "2020")) %>% slice(1)
df_initial <- as.numeric(df_initial[1,2:6])

# get the difference for the value reported in each month
#    compared with the January value of the same year
df_diff_NIC <- df_total_NIC %>%
  mutate(NIC_diff = NIC - lag(NIC) ) %>%
  melt(id = c("month", "year"), measure="NIC_diff") %>%
  dcast(month ~ year) %>%
  subset(select = c("month", "2016", "2017", "2018", "2019", "2020"))
the_months <- df_diff_NIC[[1]]
the_columns <- colnames(df_diff_NIC)
df_diff_NIC[1,2:6] <- 0
```

```r
df_diff_NIC <- df_diff_NIC[,2:6] %>%
  apply(2, cumsum) %>% matrix(ncol=5)


# get the growth dataframe
my_matrix <- sweep(df_diff_NIC, 2, df_initial, "/") * 100
df_growth_NIC <- data.frame(the_months, my_matrix) %>%
  rename_all(~the_columns) %>%
  melt(id=c("month"))

# plot
df_growth_NIC %>% ggplot(aes(x = month, y = value, color = variable, group = variable)) +
  geom_line() +
  geom_point() +
  labs(title = "Cumulative Growth of Banknotes in Circulation",
       subtitle = "2016-2020",
       x="Month", y="Percentage (%)",
       color = "Year") +
  theme_bw()
```




```r
# save plot
ggsave(paste0(my_route, "Cumulative Growth of Banknotes in Circulation.png"))
```

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