SIE example with Counterfeit Banknotes and Coins series

Victor Cuspinera

22/10/2020

This is the example shared in the **Usage** section of the main introduction of this Repository.

Overview

To show the usage of the SIE API with `siebanxicor` R-package, we will go through an example using the time series for the of the **Annual counterfeit domestic banknotes detected for the current denomination in circulation**: 20, 50, 100, 200, 500 and 1000 pesos (series **SM1249**, **SM1250**, **SM1251**, **SM1252**, **SM1253**, **SM1254**).

Click here to look for the complete catalogue of the SIE's time series published by Banco de Mexico.

1. Load library

After the `siebanxicor` package is installed, load this library and also the tidyverse library.

```
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()
```

2. Use `setToken(token)`

Bring your token and open the SIE API channel with the `setToken` utility function.

```
# bring the token
token_file <- read.csv("../token/SIE_Token.csv", header=FALSE)

# set the token
setToken(token_file$V2)
```

Notes:
- If you don't have a token to use SIE API, ***click here*** *to access the official website and obtain one.*
- I add a csv file where users should paste and save their token to run this example.

3. Get data with `getSeriesData(series, startDate, endDate)`

Get the time series of interest, in this case the series of the annual counterfeit of mexican banknotes per denomination, using the `getSeriesData` function.

```
# setting the variables
my_series <- c("SM1249", "SM1250", "SM1251", "SM1252", "SM1253", "SM1254")
my_start <- '2015-01-01'
my_end <- Sys.Date() #looks for today's date

# getting the series
series <- getSeriesData(my_series, my_start, my_end)
```

...this is the vector returned as result:

```
## $SM1249
## $SM1249$date
## [1] "2015-01-01" "2016-01-01" "2017-01-01" "2018-01-01" "2019-01-01"
##
## $SM1249$value
## [1] 512 672 296 213 311
##
##
## $SM1250
## $SM1250$date
## [1] "2015-01-01" "2016-01-01" "2017-01-01" "2018-01-01" "2019-01-01"
##
## $SM1250$value
## [1] 11291  8798  7487  6308  3080
##
##
## $SM1251
## $SM1251$date
## [1] "2015-01-01" "2016-01-01" "2017-01-01" "2018-01-01" "2019-01-01"
##
## $SM1251$value
## [1] 55614 49777 57301 75947 52749
##
##
## $SM1254
## $SM1254$date
## [1] "2015-01-01" "2016-01-01" "2017-01-01" "2018-01-01" "2019-01-01"
##
## $SM1254$value
## [1] 7162 6929 5689 4312 4270
##
##
## $SM1253
## $SM1253$date
## [1] "2015-01-01" "2016-01-01" "2017-01-01" "2018-01-01" "2019-01-01"
##
## $SM1253$value
## [1] 116968 129406 142494 156544 158477
##
```

```
##
## $SM1252
## $SM1252$date
## [1] "2015-01-01" "2016-01-01" "2017-01-01" "2018-01-01" "2019-01-01"
##
## $SM1252$value
## [1] 72825 73517 87808 96331 84043
```

Note: to use the `getSeriesData` *function, you should previously call* `setToken`*.*

4. Get the metadata with `getSeriesMetadata(series, locale)`

This function returns the general information of series. To select the language of the metadata, set the *locale* variable as "en" for English, and "es" for Spanish.

```
# getting the metadata
getSeriesMetadata(my_series, locale="en")
```

```
##   idSerie
## 1  SM1253
## 2  SM1249
## 3  SM1254
## 4  SM1251
## 5  SM1250
## 6  SM1252
##                                                                          title
## 1  Annual counterfeit notes detected per denomination (domestic banknotes), 500 pesos
## 2   Annual counterfeit notes detected per denomination (domestic banknotes), 20 pesos
## 3 Annual counterfeit notes detected per denomination (domestic banknotes), 1000 pesos
## 4  Annual counterfeit notes detected per denomination (domestic banknotes), 100 pesos
## 5   Annual counterfeit notes detected per denomination (domestic banknotes), 50 pesos
## 6  Annual counterfeit notes detected per denomination (domestic banknotes), 200 pesos
##    startDate    endDate frequency dataType   unit
## 1 2006-01-01 2019-01-01    Annual    Flows Pieces
## 2 2006-01-01 2019-01-01    Annual    Flows Pieces
## 3 2006-01-01 2019-01-01    Annual    Flows Pieces
## 4 2006-01-01 2019-01-01    Annual    Flows Pieces
## 5 2006-01-01 2019-01-01    Annual    Flows Pieces
## 6 2006-01-01 2019-01-01    Annual    Flows Pieces
```

Note: to use the `getSeriesMetadata` *function, you should previously call* `setToken`*.*

5. Get a data frame of one series using `getSerieDataFrame(series, idSerie)`

This function will be helpful to get a data frame for the annual counterfeit number of 500 pesos banknotes (**SM1253** series), from the vector returned by the `getSerieDataFrame` in the previous point #3.

```
# getting the series
df_SM1253 <- getSerieDataFrame(series, "SM1253")
```

. . . this is the data frame that we get as result:

```
##         date  value
## 1 2015-01-01 116968
## 2 2016-01-01 129406
## 3 2017-01-01 142494
## 4 2018-01-01 156544
## 5 2019-01-01 158477
```

Note: to use the `getSerieDataFrame` function, you should previously call `setToken` and `getSerieData`.

6. Get the last value of one or more series with `getSeriesCurrentValue(series)`

To get the last value of the annual counterfeit banknotes per denomination series, we will use the `getSeriesCurrentValue` function.

```
series_last <- getSeriesCurrentValue(my_series)
```

...this is the data frame taht we get as result after using `getSeriesData`:

```
series_last
```

```
##   idSerie       date  value
## 1  SM1251 2019-01-01  52749
## 2  SM1252 2019-01-01  84043
## 3  SM1253 2019-01-01 158477
## 4  SM1250 2019-01-01   3080
## 5  SM1254 2019-01-01   4270
## 6  SM1249 2019-01-01    311
```

Note: to use the `getSeriesCurrentValue` function, you should previously call `setToken`.

7. Use the custome function `SIE_function(series_codes, series_names, title_plot, x_lab, y_lab, startDate, endDate, route)`

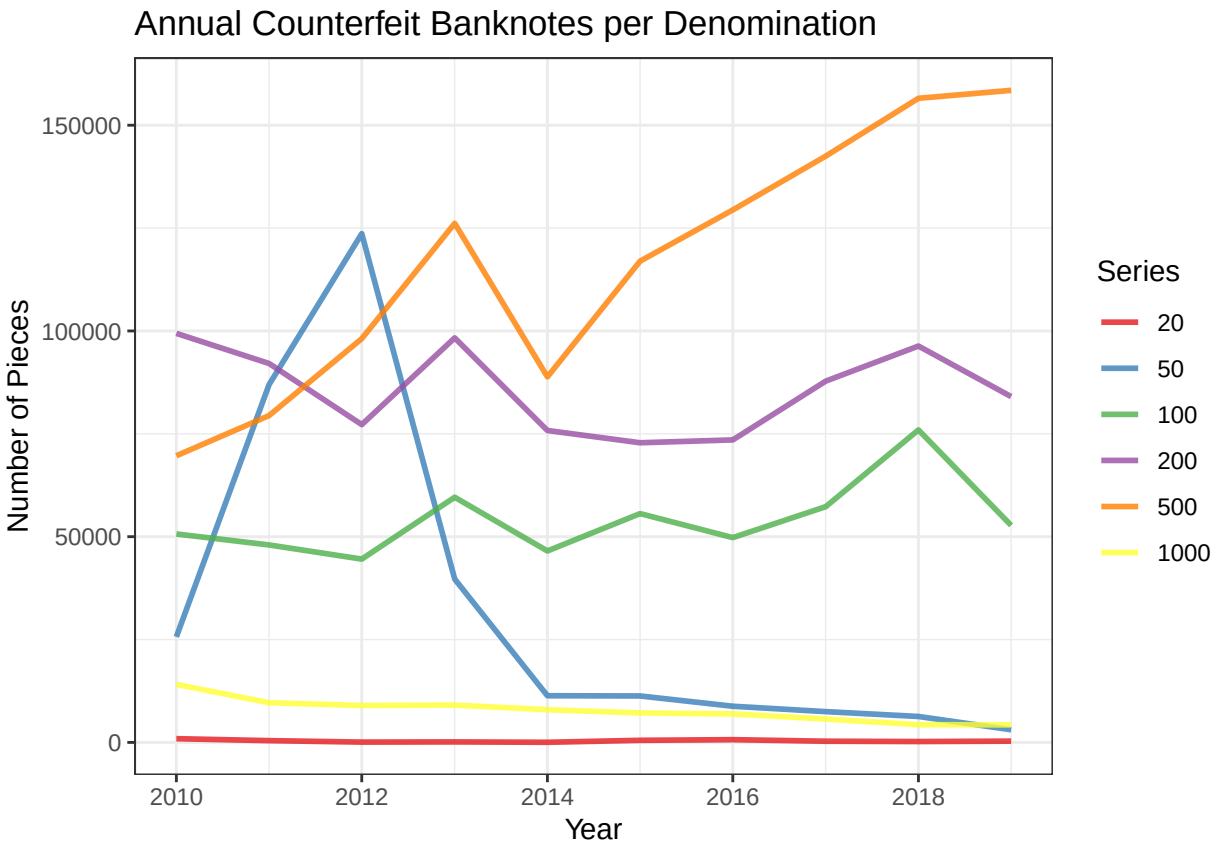
This function prints the metadata for the Annual Counterfit Banknotes series, prints and saves and their plot, and returns a data frame with these series in tidy format.

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

# setting the variables
my_series <- c("SM1249", "SM1250", "SM1251", "SM1252", "SM1253", "SM1254")
my_names <- c("20", "50", "100", "200", "500", "1000")
my_title <- "Annual Counterfeit Banknotes per Denomination"
my_start <- '2010-01-01'
my_end <- Sys.Date() #looks for today's date

# run the function
df <- sie_function(my_series, my_names, my_title, route="../img/",
             y_lab="Number of Pieces", x_lab="Year", startDate=my_start)
```

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




```
##   idSerie
## 1  SM1251
## 2  SM1252
## 3  SM1250
## 4  SM1253
## 5  SM1254
## 6  SM1249
##                                                                          title
## 1  Annual counterfeit notes detected per denomination (domestic banknotes), 100 pesos
## 2  Annual counterfeit notes detected per denomination (domestic banknotes), 200 pesos
## 3   Annual counterfeit notes detected per denomination (domestic banknotes), 50 pesos
## 4  Annual counterfeit notes detected per denomination (domestic banknotes), 500 pesos
## 5 Annual counterfeit notes detected per denomination (domestic banknotes), 1000 pesos
## 6   Annual counterfeit notes detected per denomination (domestic banknotes), 20 pesos
##    startDate    endDate frequency dataType   unit
## 1 2006-01-01 2019-01-01    Annual    Flows Pieces
## 2 2006-01-01 2019-01-01    Annual    Flows Pieces
## 3 2006-01-01 2019-01-01    Annual    Flows Pieces
## 4 2006-01-01 2019-01-01    Annual    Flows Pieces
## 5 2006-01-01 2019-01-01    Annual    Flows Pieces
## 6 2006-01-01 2019-01-01    Annual    Flows Pieces
```

. . . this is the data frame that we get as result:

```
##         date value  serie serie_name
```

```
## 1 2010-01-01   912 SM1249         20
## 2 2011-01-01   438 SM1249         20
## 3 2012-01-01    81 SM1249         20
## 4 2013-01-01   127 SM1249         20
## 5 2014-01-01    31 SM1249         20
## 6 2015-01-01   512 SM1249         20
```