27 de diciembre de 2020

ANÁLISIS DE DATOS DE TIPOS DE CAMBIO EN PYTHON

Pablo Ramos del Busto

POLRDB@GMAIL.COM

linkedin.com/in/pablo-ramosdb/

ANÁLISIS DE DATOS DE TIPOS DE CAMBIO

1. INTRODUCCIÓN

Es importante entender que este documento solo tratará de explicar el código mostrado en el artículo. Este documento tan solo será orientativo y no enseñará nada de economía.

Este documento trata de explicar de manera sencilla y breve como llevar a cabo el tratamiento de datos de mercados obtenidos a través de una API de la siguiente página, para su análisis automático a través de Python.

2. TIPOS DE CAMBIO EN PYTHON

Primero se va a explicar cómo obtener datos a través de la API ***financialmodelingprep***, que ofrece una gran variedad de datos financieros de forma gratuita, y graficar tipos de cambio con Python. Se construirá un código para recuperar el precio histórico de los tipos de cambio de varias divisas y se graficarán usando la librería ***matplotlib***.

2.1. Paso 1

Antes de empezar aconsejo crear un ***workspace*** propio en Python y un entorno virtual en donde instalar los paquetes necesarios, así como realizar los cambios de configuración que se crean convenientes. Para ello aconsejo utilizar el siguiente documento como apoyo.

Los paquetes que se van a descargar serán del repositorio ***PyPi***:

- Pandas
- Resquests
- Matplotlib

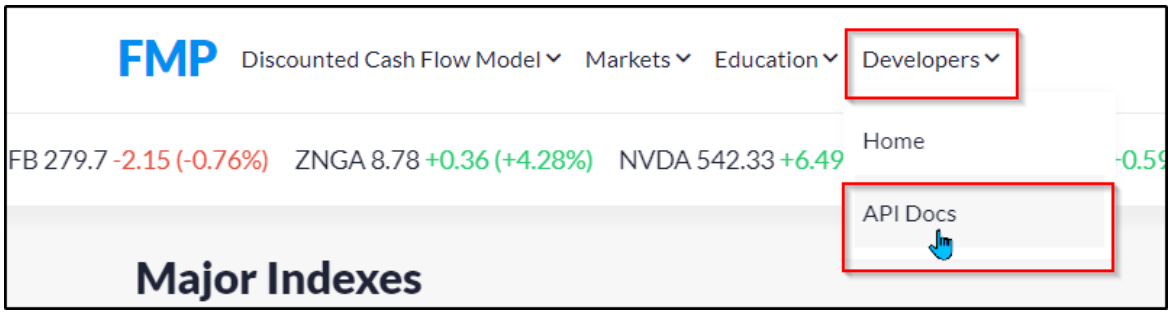
```
#Importamos los paquetes que vamos a necesitar
import requests
import pandas as pd
import matplotlib.pyplot as plt
import json
```

2.2. Paso 2

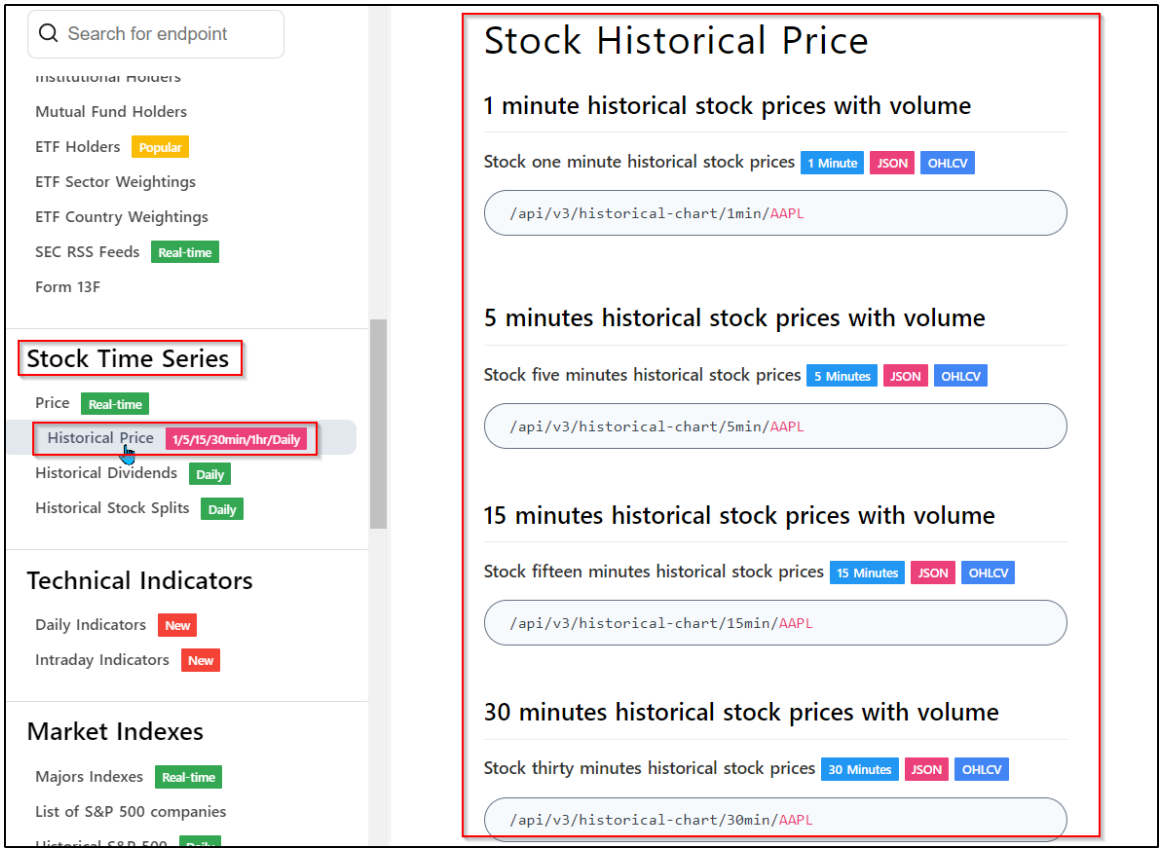
Después de importar los paquetes se realizará una llamada a la API para recuperar el histórico de los últimos años para algunos tipos de cambio. Para ello se programará un bucle donde, en cada iteración, se obtendrá el precio histórico de una divisa.

2.2.1. ENTENDER LA API

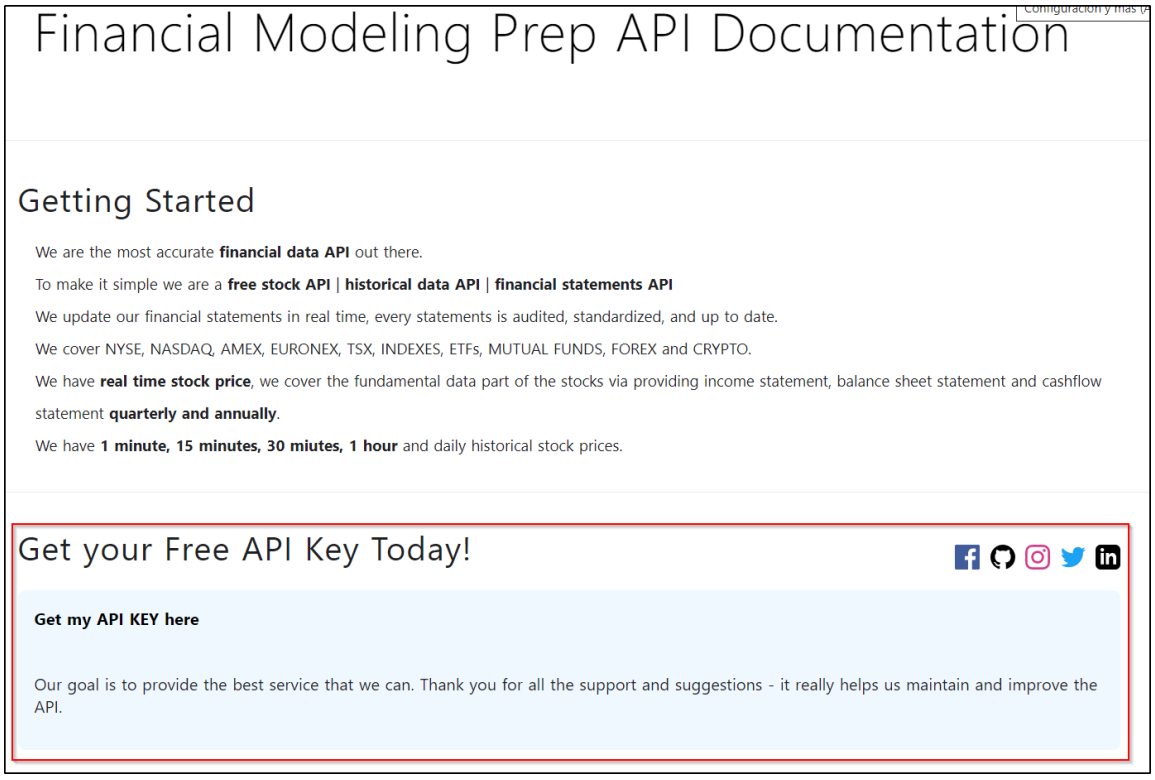
Llegados a este punto, se debe comprender el funcionamiento de la API de la página web. Para acceder a los documentos API, donde se indica como obtener los datos, se hará clic en el menú para ***developers***.



Este caso nos ocupa los precios históricos, por lo que buscamos en la lista del margen izquierdo la sección correspondiente. Una vez ahí, se explica cómo obtener diferentes medidas para esta función. En el caso que trata este documento serán los ***daily historial Price*** dentro de ***stock time series***.



Antes de empezar a extraer los valores es necesario poseer un API Key que permita extraer los valores que se desean. Para realizar esto, se debe ingresar una cuenta en la siguiente dirección.



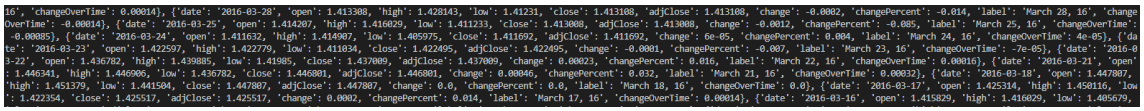
El realizar el registro, se puede seleccionar un plan gratuito que dará acceso al suficiente número de datos para realizar la prueba y valorar la compra de la base de datos. Una vez registrados se aportará un API Key el cual servirá para realizar la conexión y poder realizar las consultas deseadas.

Para realizar la llamada de datos se utiliza el método ***get*** del paquete ***request***, que almacenarán los datos recibidos en un ***json*** (los datos están definidos de este modo).

A continuación, se pasará al tratamiento del ***json*** y mostrar los resultados obtenidos en el terminal.

En la imagen superior se presenta la creación de un ***array*** con las divisas que se desea obtener. Posteriormente, se crea un ***string*** donde se almacena el código de la API que se proporcionó al realizar el registro en la web.

El bucle ***for*** servirá para ir recorriendo cada valor del array ***currencies***. Los datos obtenidos presentan un formato ***json***, como bien se especifica en la web. Así es cómo se verían los datos impresos en pantalla:



A simple vista se puede apreciar una barbaridad de datos sin sentido y orden alguno, que necesitan ser tratados para hacerlos entendibles para el usuario.

2.3. Paso 3

Lo que se debe realizar ahora es el tratamiento de los datos, para obtener aquello que se considere de valor. Para ello, la propia web permite ver la forma que tiene el ***json*** que se devuelve, como se observa en la imagen inferior.

Posteriormente, en la variable ***forexJSON***, se recorre por cada divisa todo el histórico (***historical***). Se almacenan dos variables: el precio de cierre con su fecha correspondiente. Esto se almacenará en el ***dict Exchange_rates_Python[currency]***, donde ***currency*** equivale a la divisa. Todo este proceso se puede ver en la imagen inferior.

```
#Almacenamos en el array los valores deseados
exchange_rates_Python[currency] = {}

for item in forexJSON['historical']:
    adj_close = item['adjClose']
    trade_date = item['date']
    exchange_rates_Python[currency][trade_date] = adj_close
```

Si se desean realizar algunas pruebas tan solo añadiendo al final del ***for*** una de las divisas que está en la lista ***currencies*** y una de las fechas que se pueden ver en el ***json*** se puede ver el formato.

```
print(exchange_rates_Python[divisa][fecha])
```

```
21      #Almacenamos en un dict los valores deseados
22      exchange_rates_Python[currency] = {}
23
24      for item in forexJSON['historical']:
25          adj_close = item['adjClose']
26          trade_date = item['date']
27          exchange_rates_Python[currency][trade_date] = adj_close
28
29  print(exchange_rates_Python['EURUSD']['2020-12-03'])
30
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
1.21146
(.venv) PS C:\PoryectosPython\2020-12-07\Impact_Of_Exchange_Rates_In_Companies>
```

2.4. Paso 4

Posteriormente lo que se realiza es la conversión de los datos al tipo ***DataFrame*** usando el paquete ***Pandas***. Como los datos están en forma ***dictionary*** se debe utilizar la función ***from_dict***.

	EURUSD	CHFUSD=X	AUDUSD	GBPUSD
2020-12-08	1.211974	1.122536	0.742710	1.335952
2020-12-07	1.212136	1.121403	0.743157	1.341742
2020-12-04	1.214565	1.122624	0.743898	1.345497
2020-12-03	1.211460	1.117805	0.741801	1.337077
2020-12-02	1.207438	1.111716	0.738089	1.342264
...	...	...	...	...
2015-12-14	1.098394	1.016467	0.716486	1.519896
2015-12-11	1.094811	1.012146	0.727220	1.516093
2015-12-10	1.101795	1.016984	0.723537	1.518395
2015-12-09	1.089396	1.007252	0.722507	1.501006
2015-12-08	1.083400	0.999600	0.725900	1.505027

Por ultimo y como se ve en la imagen superior se convierten los valores índices, es decir, los valores “cabecera” de las filas en formato tiempo por medio de la función ***to_datetime***.

```
#Salimos del bucle inicial de currencies y convertimos los datos en DataFrame
currenciesDataFrame = pd.DataFrame.from_dict(exchange_rates_Python)

#Convertimos los valores "cabeceros" de las filas a formato tiempo
currenciesDataFrame.index = pd.to_datetime(currenciesDataFrame.index)
```

2.5. Paso 5

Para acabar y representar por medio de figuras los valores, se utilizará el paquete ***matplotlib***, importado al inicio con el resto de paquetes.

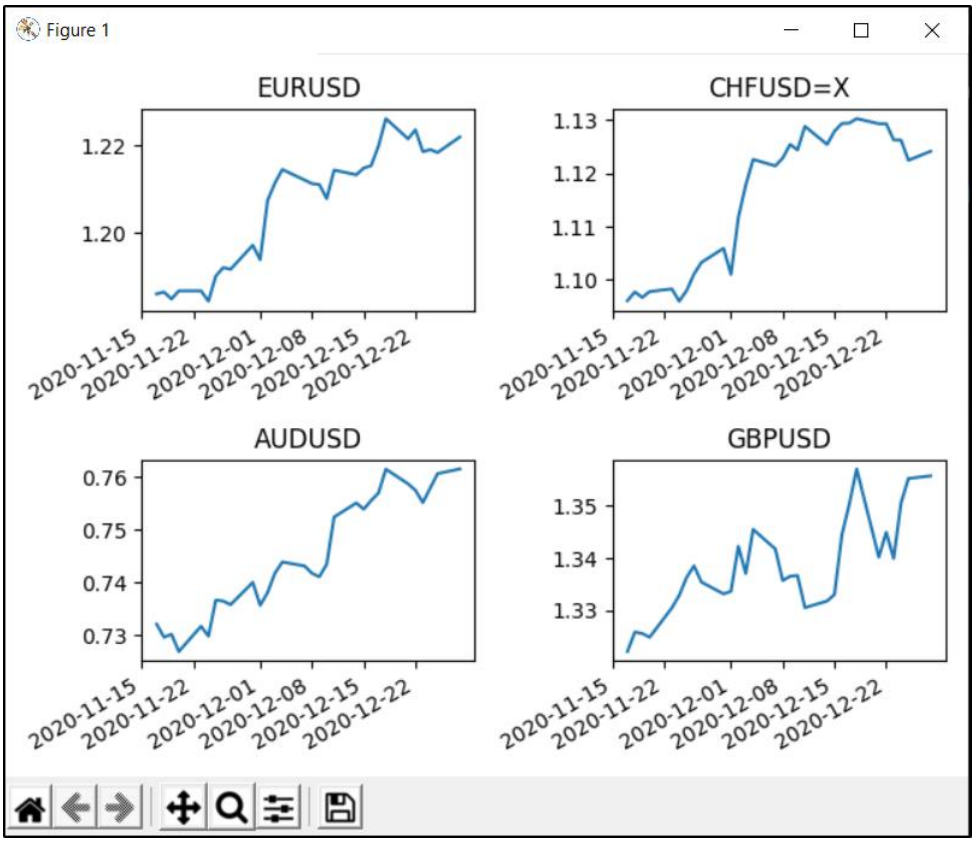
En la imagen representada inferiormente se seleccionarán con la función ***iloc*** los valores de los últimos 30 días. Esto se debe a que se pone el valor después de los ‘:’ (desde : hasta) de esta manera indicamos hasta donde. Por el contrario, si se llega a poner el valor previo a los ‘:’ se indicaría desde donde.

Se definen el número de gráficos que se representarán en la misma figura por medio de la función ***subplot***. Posteriormente, se van creando las figuras por cada tipo de divisa y situando esta en la posición deseada, dentro de la figura creada con la función ***subplot().***

```
#Representamos las figuras
fig, axes = plt.subplots(nrows=2, ncols=2)

currenciesDataFrameLastWeek[currencies[0]].plot(ax=axes[0,0])
axes[0,0].set_title(currencies[0])

currenciesDataFrameLastWeek[currencies[1]].plot(ax=axes[0,1])
axes[0,1].set_title(currencies[1])

currenciesDataFrameLastWeek[currencies[2]].plot(ax=axes[1,0])
axes[1,0].set_title(currencies[2])

currenciesDataFrameLastWeek[currencies[3]].plot(ax=axes[1,1])
axes[1,1].set_title(currencies[3])

plt.tight_layout()
plt.show()
```

Filas, columnas

Una vez escrito adecuadamente todo el código con las función ***show()*** se representarán las figuras en la forma definida. Serán una figura con cuatro gráficos, uno por cada divisa.



3. RECURSOS

Para la realización de este documento se utilizó como recursos las siguientes páginas web:

- Impact of exchange rates in companies | by Jose Manu (CodingFun) | Towards Data Science
- Financial Modeling Prep - Home
- PyPI · The Python Package Index
- Python Tutorial (w3schools.com)
- pandas documentation — pandas 1.2.0 documentation (pydata.org)

Además, si se desea tener acceso completo al código se adjunta el link de este.

An-lisis-de-datos-de-tipos-de-cambio/impactOfExchangeRatesInCompanies at main · PolBranch5/An-lisis-de-datos-de-tipos-de-cambio (github.com)