Petroineos Summer Internship

Fayeb Rahman

June 2024

Comments

Below I go over my thought process behind the program which I created for the task at hand. Lines of code are notoriously hard to understand when looking at them for the first time so hopefully this document should help outline the key steps which I took to solve the problem.

load-new-data-from-file

In this function, all I did was import the given csv files. Since this raw data was not yet in a readily usable form, I stored this information in a dummy variable 'new-stuff'.

save-new-data

This function makes up the majority of the code. I list below the key bits of this function.

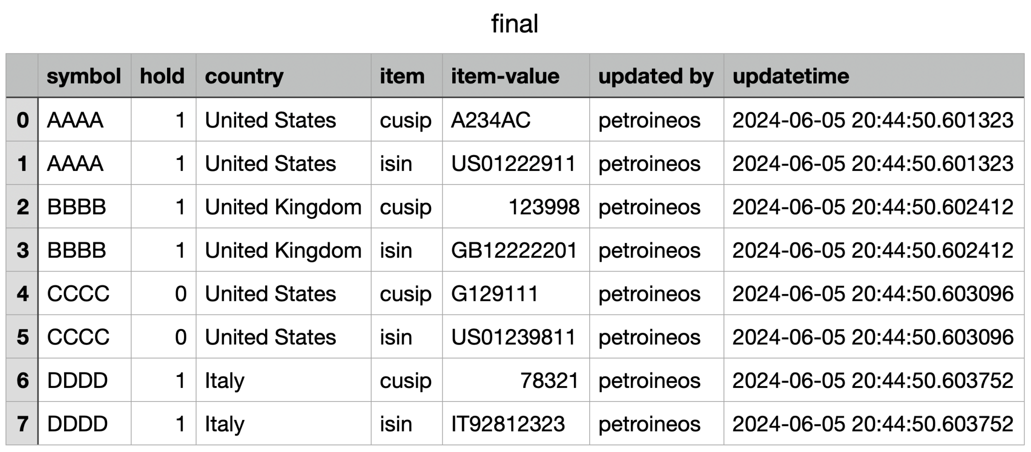
- First, I sift through the current database to see which symbols I already have. I put all of these 'symbols' in a list.
- Now I look at the new pieces of data which we receive via the function which loads the new data. Here, there are two main cases to consider. If a new row of data has the same symbol as in our current database, I just ammended the current database row to include the new pieces of information. This just involves iterating through a list and correct use of the 'index' function. If however we get a new symbol, i.e. 'GGGG' does not appear in the first symbols-update but does in the second, I just append this new information to the current 'database' panda dataframe.
- All the data from the csv files could go straight into the 'database' dataframe without corrections. The only exceptions to this were the 'country' and 'updatetime' columns. Here, we just had to use the packages **pycountry** and **datetime** to get the relevant information out of the data.

get-data-from-database

In this function, all we do is return the current, most recently updated, database file.

Output of the Function

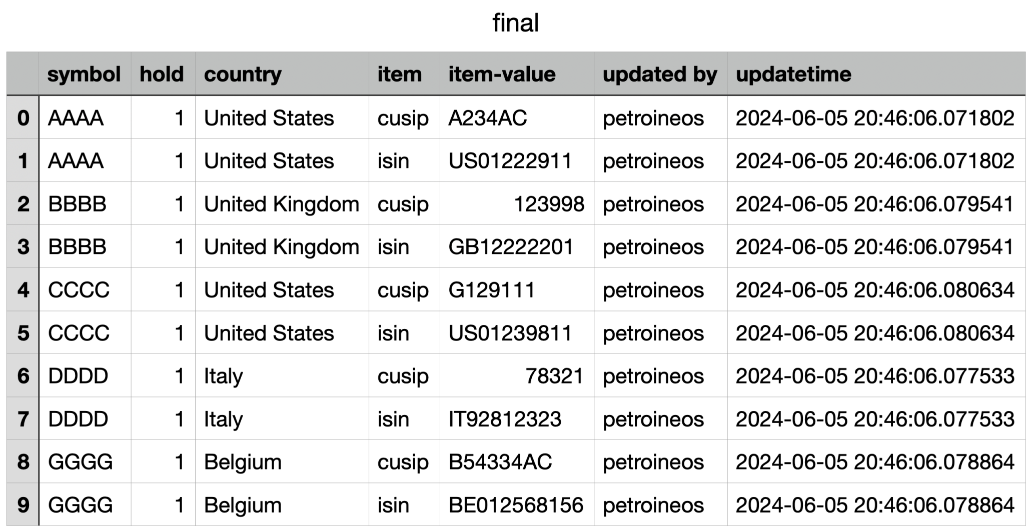
In order to see if my function was working properly, I printed the dataframe out after every update. The print-outs can be seen below:



final

	symbol	hold	country	item	item-value	updated by	updatetime
0	AAAA	1	United States	cusip	A234AC	petroineos	2024-06-05 20:44:50.601323
1	AAAA	1	United States	isin	US01222911	petroineos	2024-06-05 20:44:50.601323
2	BBBB	1	United Kingdom	cusip	123998	petroineos	2024-06-05 20:44:50.602412
3	BBBB	1	United Kingdom	isin	GB12222201	petroineos	2024-06-05 20:44:50.602412
4	CCCC	0	United States	cusip	G129111	petroineos	2024-06-05 20:44:50.603096
5	CCCC	0	United States	isin	US01239811	petroineos	2024-06-05 20:44:50.603096
6	DDDD	1	Italy	cusip	78321	petroineos	2024-06-05 20:44:50.603752
7	DDDD	1	Italy	isin	IT92812323	petroineos	2024-06-05 20:44:50.603752

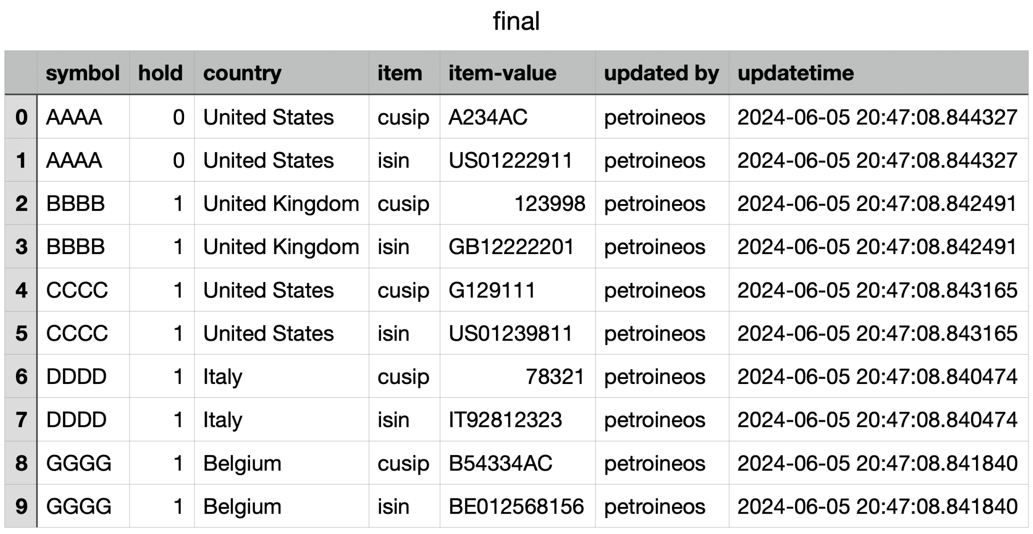
Figure 1: The Symbols Update dataframe after the first update.



final

	symbol	hold	country	item	item-value	updated by	updatetime
0	AAAA	1	United States	cusip	A234AC	petroineos	2024-06-05 20:46:06.071802
1	AAAA	1	United States	isin	US01222911	petroineos	2024-06-05 20:46:06.071802
2	BBBB	1	United Kingdom	cusip	123998	petroineos	2024-06-05 20:46:06.079541
3	BBBB	1	United Kingdom	isin	GB12222201	petroineos	2024-06-05 20:46:06.079541
4	CCCC	1	United States	cusip	G129111	petroineos	2024-06-05 20:46:06.080634
5	CCCC	1	United States	isin	US01239811	petroineos	2024-06-05 20:46:06.080634
6	DDDD	1	Italy	cusip	78321	petroineos	2024-06-05 20:46:06.077533
7	DDDD	1	Italy	isin	IT92812323	petroineos	2024-06-05 20:46:06.077533
8	GGGG	1	Belgium	cusip	B54334AC	petroineos	2024-06-05 20:46:06.078864
9	GGGG	1	Belgium	isin	BE012568156	petroineos	2024-06-05 20:46:06.078864

Figure 2: The Symbols Update dataframe after the second update.



final

	symbol	hold	country	item	item-value	updated by	updatetime
0	AAAA	0	United States	cusip	A234AC	petroineos	2024-06-05 20:47:08.844327
1	AAAA	0	United States	isin	US01222911	petroineos	2024-06-05 20:47:08.844327
2	BBBB	1	United Kingdom	cusip	123998	petroineos	2024-06-05 20:47:08.842491
3	BBBB	1	United Kingdom	isin	GB12222201	petroineos	2024-06-05 20:47:08.842491
4	CCCC	1	United States	cusip	G129111	petroineos	2024-06-05 20:47:08.843165
5	CCCC	1	United States	isin	US01239811	petroineos	2024-06-05 20:47:08.843165
6	DDDD	1	Italy	cusip	78321	petroineos	2024-06-05 20:47:08.840474
7	DDDD	1	Italy	isin	IT92812323	petroineos	2024-06-05 20:47:08.840474
8	GGGG	1	Belgium	cusip	B54334AC	petroineos	2024-06-05 20:47:08.841840
9	GGGG	1	Belgium	isin	BE012568156	petroineos	2024-06-05 20:47:08.841840

Figure 3: The Symbols Update dataframe after the final update.

As a sanity check, we can see that, after the final update, the dataframe looks exactly like it should! For closure, I include the actual program below too.

Thank you,
Fayeb Rahman.

Program

```
import pandas as pd
import pycountry
from datetime import datetime


class SymbolsUpdate(object):
    def __init__(self):
        self.new_stuff = 'new.csv'
        self.database_file = 'database.csv'
        stuff = ['symbol', 'hold', 'country', 'item', 'item-value', 'updated by', '
                                                    updatetime']
        self.database = pd.DataFrame(columns=stuff)

    def load_new_data_from_file(self, file_path: str):
        self.new_stuff = pd.read_csv(file_path)
        return self.new_stuff
        pass

    def save_new_data(self, input_data: pd.DataFrame):
        x = self.database.symbol
        old = []
        for j in range(0, len(x)):
            old.append(x[j])

        y = input_data.symbol

        for i in range(0, len(y)):
            new_symbol = y[i]
            n_r = input_data.iloc[i]
            country = pycountry.countries.get(alpha_2=str(n_r['isin'])[:2])
            time = datetime.now()
            item_1 = [n_r['symbol'], n_r['hold'], country.name, 'cusip', n_r['cusip']
                                                        , 'petroineos', time]
            item_2 = [n_r['symbol'], n_r['hold'], country.name, 'isin', n_r['isin'],
                                                      'petroineos', time]
            if new_symbol in old:
                j = old.index(new_symbol)
                self.database.loc[j] = item_1
                self.database.loc[j+1] = item_2
            if new_symbol not in old:
                self.database.loc[len(x) + 2*i] = item_1
                self.database.loc[len(x) + 2*i + 1] = item_2

        pass

    def get_data_from_database(self):
        self.database_file = self.database.sort_values(by='symbol')
        return self.database_file
        pass


su = SymbolsUpdate()
new_data = su.load_new_data_from_file('symbols_update_1.csv')
su.save_new_data(new_data)


new_data = su.load_new_data_from_file('symbols_update_2.csv')
su.save_new_data(new_data)


new_data = su.load_new_data_from_file('symbols_update_3.csv')
su.save_new_data(new_data)


output = su.get_data_from_database()

output.to_csv('final.csv')
```