Bài 1:

1. Collect data cho toàn bộ 30 mã cổ phiếu trong một ngày. (code cần chạy được live)

   Link code: https://github.com/jaxphuchanlmht1999/crawl_trading_data_from_API_080323

2. Tìm ra xem trường nào là trường total volume, dùng trường đó để thiết lập lại thứ tự data trả về

   Total value là trường "va"

3. Dùng trường total volume để tính xem có data-point nào bị thiếu không

   Code check data-point bị thiếu dựa trên trường total_volume:

```
for i in range(len(df['va']) - 1):
    if(int(df['va'][i:i+1]) == int(df['va'][i+1:i+2])):
        print(df[df['va'] == int(df['va'][i:i+1])])
```

[31] ✓ 0.0s

```
...          o      t       c   ch       h       l   mb   mv    ra   se  \
74  105500  71800  103800 -700  105500  103800  ASK  200 -0.67  241
75  105500  71800  103800 -700  105500  103800  ASK  100 -0.67  240

             va     vo
74  6634610000  63300
75  6634610000  63300
         o      t       c   ch       h       l   mb   mv    ra   se  \
81  105500  71611  104000 -500  105500  104000  ASK  100 -0.48  234
82  105500  71611  104000 -500  105500  104000  ASK  100 -0.48  234

             va     vo
81  6541060000  62400
82  6541060000  62400
         o      t       c   ch       h       l   mb   mv    ra   se  \
83  105500  71611  104000 -500  105500  104000  ASK  100 -0.48  232
84  105500  71611  104000 -500  105500  104000  ASK  200 -0.48  231

             va     vo
83  6520260000  62200
84  6520260000  62200
         o      t       c   ch       h       l   mb   mv    ra   se  \
92  105500  71450  104000 -500  105500  104000  ASK  100 -0.48  223
93  105500  71450  104000 -500  105500  104000  ASK  100 -0.48  223

             va     vo
92  5958660000  56800
93  5958660000  56800
```

4. Dựa vào data có được, tính thanh khoản của vn30 trong từng giờ một

Thanh khoản = df[‘mv’]*df[‘vo’]

```
1  df['thanh_khoan'] = df['mv'] * df['vo']
2  df[:10]
```

✓ 0.0s

	o	t	c	ch	h	l	mb	mv	ra	se	va	vo	thanh_khoan
70	105500	72020	103900	-600	105500	103800	BID	200	-0.57	245	6717700000	64100	12820000
71	105500	72011	103900	-600	105500	103800	BID	100	-0.57	244	6707310000	64000	6400000
72	105500	72006	103900	-600	105500	103800	BID	100	-0.57	243	6696920000	63900	6390000
73	105500	72000	103900	-600	105500	103800	BID	300	-0.57	242	6665750000	63600	19080000
74	105500	71800	103800	-700	105500	103800	ASK	200	-0.67	241	6634610000	63300	12660000
75	105500	71800	103800	-700	105500	103800	ASK	100	-0.67	240	6634610000	63300	6330000
76	105500	71748	103900	-600	105500	103800	BID	100	-0.57	239	6624220000	63200	6320000
77	105500	71619	103800	-700	105500	103800	ASK	100	-0.67	238	6613840000	63100	6310000
78	105500	71619	103900	-600	105500	103900	ASK	200	-0.57	237	6593060000	62900	12580000
79	105500	71616	104000	-500	105500	104000	ASK	100	-0.48	236	6582660000	62800	6280000

5. Sau khi collect data vn30 được vài ngày, thống kê lại xem giờ giao dịch lúc nào là thanh khoản cao nhất

Sau khi thu thập dữ liệu VN_30 từ 3 ngày 5-7/03/2023, thời gian thu thập từ 9h30-16h30, cách 1 tiếng thu thập 1 lần , sau khi nối các dataframe các ngày lại, và lấy top 20 các dòng dữ liệu có tính thanh khoản cao nhất, ta nhận thấy rằng, trong khoảng thời gian(72853,71533), tức là 14h15p đến 14h28p sẽ là khoảng thời gian mà có tính thanh khoản cao nhất.

	t	c	ch	h	l	mv	ra	se	va	vo	thanh_khoan
553	71753	1019.28	5.93	1019.28	1019.28	1037400	0.59	2332.0	2191192000000	96966900	100593462060000
581	71533	1019.44	6.09	1019.44	1019.44	673200	0.60	2304.0	2131939000000	93976800	63265181760000
506	72149	1017.99	4.64	1017.99	1017.99	566100	0.46	2379.0	2265127000000	100011900	56616736590000
503	72853	1024.33	10.07	1024.33	1024.33	356700	0.99	2461.0	3073385000000	136591500	48722188050000
511	72813	1026.72	12.46	1026.72	1026.72	291800	1.23	2453.0	3048202000000	135450500	39524455900000
521	72722	1027.43	13.17	1027.43	1027.43	288500	1.30	2443.0	3014456000000	133947700	38643911450000
578	72237	1028.42	14.16	1028.42	1028.42	296300	1.40	2386.0	2903766000000	129347900	38325782770000
570	72317	1027.17	12.91	1027.17	1027.17	288500	1.27	2394.0	2928209000000	130359500	37608715750000
513	72802	1026.66	12.40	1026.66	1026.66	272500	1.22	2451.0	3042159000000	135095300	36813469250000
533	72622	1028.12	13.86	1028.12	1028.12	263600	1.37	2431.0	2980946000000	132506300	34928660680000
515	72752	1026.13	11.87	1026.13	1026.13	257500	1.17	2449.0	3035352000000	134748400	34697713000000
572	71618	1018.08	4.73	1018.08	1018.08	354700	0.47	2313.0	2160889000000	95178600	33759849420000
518	72737	1027.23	12.97	1027.23	1027.23	240500	1.28	2446.0	3026797000000	134399100	32322983550000
524	72707	1027.75	13.49	1027.75	1027.75	241200	1.33	2440.0	3004233000000	133489200	32197595040000
534	72617	1029.04	14.78	1029.04	1029.04	224700	1.46	2430.0	2975042000000	132242700	29714934690000
522	72029	1018.28	4.93	1018.28	1018.28	301900	0.49	2363.0	2225015000000	98330100	29685857190000
506	72838	1025.93	11.67	1025.93	1025.93	206300	1.15	2458.0	3064042000000	136111900	28079884970000

```
1  df_top30_thanh_khoan['t'].max(), df_top30_thanh_khoan['t'].min()
```

✓ 0.0s

```
(72853, 71533)
```

```
1  import pandas as pd
2  import glob
3  import os
4
5  path = r'C:/Users/Admin/OneDrive_Quan/Máy tính/Test_CTY_Dang/Request/persistent_060323/VN30'
6  all_files = glob.glob(os.path.join(path, "*.csv"))
7
8  df_VN30_060323 = pd.concat((pd.read_csv(f) for f in all_files), ignore_index=True)
9  df_VN30_060323['thanh_khoan'] = df_VN30_060323['mv'] * df_VN30_060323['vo']
10
11 df_VN30_060323[:10]
```

✓ 0.0s

	t	c	ch	h	l	mv	ra	se	va	vo	thanh_khoan
0	22103	1023.40	10.05	1023.40	1023.40	114700	0.99	67.0	168847000000	10723100	1229939570000
1	22058	1023.34	9.99	1023.34	1023.34	22700	0.99	66.0	166017000000	10608400	240810680000
2	22053	1023.45	10.10	1023.45	1023.45	16300	1.00	65.0	165340000000	10585700	172546910000
3	22048	1023.43	10.08	1023.43	1023.43	85900	0.99	64.0	164516000000	10569400	907911460000
4	22043	1023.65	10.30	1023.65	1023.65	20400	1.02	63.0	162590000000	10483500	213863400000
5	22038	1023.65	10.30	1023.65	1023.65	158200	1.02	62.0	162091000000	10463100	1655262420000
6	22033	1023.85	10.50	1023.85	1023.85	5800	1.04	61.0	158093000000	10304900	59768420000
7	22028	1023.43	10.08	1023.43	1023.43	272200	0.99	60.0	157982000000	10299100	2803415020000
8	22023	1023.48	10.13	1023.48	1023.48	68600	1.00	59.0	153356000000	10026900	687845340000
9	22018	1023.48	10.13	1023.48	1023.48	27200	1.00	58.0	151578000000	9958300	270865760000

6. Dùng linear regression để tính xem giá cổ phiếu nào tăng giảm thì có ảnh hưởng nhiều nhất đến vn30

0 : giá cổ phiếu giảm so với lần trước

1 : giá cổ phiếu không thay đổi so với lần trước

2 : giá cổ phiếu tăng so với lần trước

	index	value
45	PDR_2	0.000001
58	SSI_0	0.000026
25	HPG_0	0.000036
12	CTG_2	0.000069
43	PDR_0	0.000086
42	NVL_2	0.000093
60	SSI_2	0.000145
28	KDH_0	0.000157
67	TPB_0	0.002263
37	MWG_0	0.004050
7	BVH_0	0.009422
87	VPB_2	0.009898
73	VHM_0	0.014638
90	VRE_2	0.021857
63	STB_2	0.027944
81	VJC_2	0.043881
27	HPG_2	0.047552

- Ở cột index: + nếu có _0 phía sau mã cổ phiếu, nghĩa là giá cổ phiếu giảm giá trị so với lần lấy trước

  + nếu có _2 phía sau mã cổ phiếu, nghĩa là giá cổ phiếu tăng giá trị so với lần lấy trước

- Ở cột Value: + giá trị càng nhỏ, thì độ ảnh hưởng tới giá của mã cổ phiếu VN_30 càng lớn

  VD: *ở index = 'PDR_2' => khi giá cổ phiếu PDR tăng, thì sẽ có ảnh hưởng lớn đến VN_30

  *ở index = 'SSI_0' => khi giá cổ phiếu SSI giảm, thì sẽ có ảnh hưởng lớn đến VN_30

7. *Code ra một thuật toán có input là giá 30 mã cổ phiếu, và output là vn30_index. Forward test thuật toán đó và thống kê xem accuracy ở mức nào*

- Tập dữ liệu gồm 1238 dòng, chia thành 2 tập train test theo tỉ lệ 0.8 / 0.2. Input là 30 mã cổ phiếu, out_put và giá của vn30.
- Mô hình và đánh giá độ chính xác được mô tả dưới bảng này

Model	R^2	MAE	MSE	RMSE
Linear Regression	0.9981	1121769	2262570426280	1504184
Logistic Regression	0.9999	52360	77440188787	278280

- Link git: https://github.com/jaxphuchanlmht1999/crawl_trading_data_from_API_080323/blob/main/Bai_1.ipynb