

Home (<http://www.patanasongsivilai.com/blog>)

/ AI/Machine Learning (<http://www.patanasongsivilai.com/blog/category/aimachine-learning/>)

/ วิธีเขียนโค้ดดึงข้อมูลหุ้นไทย ด้วยภาษา Python (แจกโค้ดฟรี)

วิธีเขียนโค้ดดึงข้อมูลหุ้นไทย ด้วยภาษา Python (แจกโค้ดฟรี)

วิธีเขียนโค้ดดึงข้อมูลหุ้นไทย ด้วยภาษา Python (แจกโค้ดฟรี)

📅 18 มิถุนายน 2017 (<http://www.patanasongsivilai.com/blog/2017/06/18/>)

👤 แอดมินโฮ ไอ้น้อยออก (<http://www.patanasongsivilai.com/blog/author/>)

📁 AI/Machine Learning (<http://www.patanasongsivilai.com/blog/category/aimachine-learning/>)

💬 ()6 comments (<http://www.patanasongsivilai.com/blog/stock-thai-python/#comments>)

บทความนี้จะกล่าวถึง ตัวอย่างเขียนภาษา **Python 3** เพื่อดึงข้อมูลจากตลาดหุ้นไทย ซึ่งมีหลากหลายวิธีดังนี้ (ผู้เขียนสรุปค่อนข้างสั้น ดังนั้นควรมีพื้นฐาน python จะดีมาก)

คำเตือน

- ถ้าเปิดบ่นถือ อาจเห็นโค้ดเพี้ยนได้ แนะนำให้เปิดบนคอมดีกว่า
- ถ้าจะลง Python เพื่อรันโค้ดตัวอย่าง ให้ติดตั้ง Anaconda (<https://www.continuum.io/downloads>) ดีกว่า (มันจะติดตั้งทั้งคอมไพเลอร์ มอดูล แพ็กเกจ IDE, IPython และอื่นๆ ที่จำเป็นสำหรับเขียน Python ให้เสร็จสรรพ)

วิธีที่ 1 ใช้โมดูล googlefinance

ติดตั้งโมดูลด้วยคำสั่ง

```
pip install googlefinance
```

ตัวอย่างโค้ด ดูข้อมูลหุ้น PTT

```
from googlefinance import getQuotes
import json
symbol = 'PTT'
print(json.dumps(getQuotes('SET:' + symbol), indent=2))
```

ได้ผลลัพธ์

```
[
  {
    "ID": "1079066631482057",
    "StockSymbol": "PTT",
    "Index": "BKK",
    "LastTradePrice": "382.00",
    "LastTradeWithCurrency": "THB382.00",
    "LastTradeTime": "12:29PM GMT+7",
    "LastTradeDateTime": "2017-06-19T12:29:33Z",
    "LastTradeDateTimeLong": "Jun 19, 12:29PM GMT+7"
  }
]
```

หมายเหตุ ชื่อย่อหุ้นไทยจะมีชื่อ “SET:” นำหน้าชื่อย่อหุ้นนั้นๆ (ค่าอาร์กิวเมนต์ตอนเรียกใช้ฟังก์ชัน getQuotes())

เช่น “SET:PTT”, “SET:AOT”, “SET:SCC” หรือถ้าจะดู SET ก็ระบุไปว่า “SET:SET”

อ่านเพิ่ม <https://pypi.python.org/pypi/googlefinance> (<https://pypi.python.org/pypi/googlefinance>)

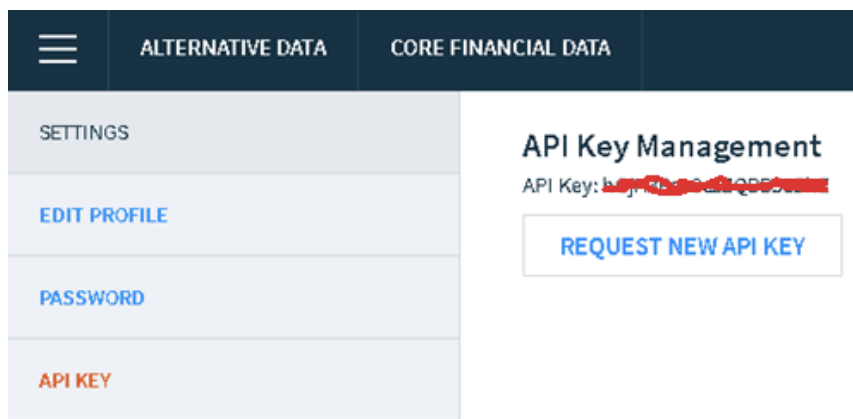
วิธีที่ 2 ใช้โมดูล quandl

ติดตั้งโมดูลด้วยคำสั่ง

```
pip install quandl
```

ไปที่เว็บ <https://www.quandl.com/> (<https://www.quandl.com/>) แล้วสมัครเป็นสมาชิกเพื่อขอ API KEY แล้วจะมีเมลส่งมาให้เรายืนยัน

หลังจากกดยืนยันเสร็จ เมื่อไปที่หน้า <https://www.quandl.com/account/api>
(<https://www.quandl.com/account/api>) ก็จะมีหน้าบอก API KEY ว่าของเราคืออะไร ดังรูปข้างล่าง



ตัวอย่างโค้ด

```
import quandl
quandl.ApiConfig.api_key = 'YOUR_API_KEY'
data = quandl.get("THAISE/INDEX")
print(data.head())
```

ในโค้ดข้างบนจะเห็นชื่อตัวแปร `quandl.ApiConfig.api_key` ก็ให้ใส่ API KEY ที่เราได้มาลงไป

ผลลัพธ์ที่ได้

	Value
Date	
2009-12-31	734.54
2010-12-31	1032.76
2011-12-31	1025.32
2012-12-31	1391.93
2013-12-31	1298.71

แต่น่าเสียดายข้อมูลมีเฉพาะดัชนี SET อย่างเดียว ข้อมูลเฉพาะหุ้นรายตัวของไทยไม่มี (หรือมีเปล่าหว่า ถ้าใครรู้ช่วยบอกที)

อ่านเพิ่ม

- <https://github.com/quandl/quandl-python> (<https://github.com/quandl/quandl-python>)
- <https://docs.quandl.com/docs/installation-1> (<https://docs.quandl.com>)

~~/docs/installation-1)~~

วิธีที่ 3 ใช้โมดูล pandas-datareader

ติดตั้งโมดูลด้วยคำสั่ง

```
pip install pandas-datareader
```

แต่ช่วงที่ผู้เขียนลองเล่นโมดูลนี้ดู ดันมีบั๊ก เลยต้องติดตั้งโมดูลข้างล่างนี้ด้วย (อนาคตถ้ามันแก้บั๊กไปแล้ว อาจไม่ต้องติดตั้งก็ได้ ที่มา <https://github.com/ranaroussi/fix-yahoo-finance> (<https://github.com/ranaroussi/fix-yahoo-finance>))

```
pip install fix_yahoo_finance --upgrade --no-cache-dir
```

ตัวอย่างโค้ด

```
from pandas_datareader import data as pdr
import fix_yahoo_finance as yf # <== that's all it takes :-)
# download dataframe
ptt = pdr.get_data_yahoo("PTT.BK", start="2017-01-01", end="2017-04-30")
print(ptt.tail())
```

หมายเหตุ ชื่อย่อหุ้นไทยจะมีชื่อ “.BK” ตามหลังชื่อย่อหุ้นนั้นๆ

เช่น “PTT.BK”, “ AOT.BK”, “SCC.BK” (แต่ผมลองแล้ว ถ้าระบุ “SET.BK” ยังไม่ได้นะครับ)

ได้ผลลัพธ์

	Open	High	Low	Close	Adj Close	Volume
Date						
2017-04-20	394.0	395.0	391.0	392.0	392.0	3020300
2017-04-21	393.0	394.0	390.0	390.0	390.0	3032300
2017-04-24	393.0	393.0	389.0	389.0	389.0	2994400
2017-04-27	387.0	391.0	386.0	388.0	388.0	2104300
2017-04-28	391.0	392.0	389.0	389.0	389.0	2451100

สังเกตวิธีนี้ให้ดีๆ ผลลัพธ์ที่แสดงออกมา จะมีราคาปิดของหุ้นอยู่ 2 ค่าได้แก่ Close (ราคาปิดธรรมดา) กับ Adj Close (Adjusted Closing Price (http://www.investopedia.com/terms/a/adjusted_closing_price.asp))

อ่านเพิ่มเติม

- <https://github.com/pydata/pandas-datareader> (<https://github.com/pydata/pandas-datareader>)
- <https://pandas-datareader.readthedocs.io/en/latest/> (<https://pandas-datareader.readthedocs.io/en/latest/>)

สำหรับวิธีที่ 1-3 ก็ดูโค้ดฉบับเต็มได้ที่

- <https://github.com/adminho/trading-stock-thailand/blob/master/datasets/miscellaneous.py> (<https://github.com/adminho/trading-stock-thailand/blob/master/datasets/miscellaneous.py>)

*** ถ้าลองรันแล้วเห็น error ไม่ต้องตกใจ ...เพราะวิธีที่ 2 ใช้โมดูล quandl ซึ่งต้องการค่า API KEY ของจริงครับ (คอมเมนต์ออกก่อนก็ได้ครับ)

วิธีที่ 4

ดาวน์โหลดไฟล์ EOD จาก <http://siamchart.com/stock/> (<http://siamchart.com/stock/>)

เมื่อไปที่เว็บ เราก็ต้องสมัครเป็นสมาชิกเว็บเขาก่อน ถึงจะโหลดได้

(เป็นวิธีลูกครึ่งนะ ครึ่งหนึ่งใช้มือดาวน์โหลด อีกครึ่งเขียนโปรแกรมเพื่อเตรียมข้อมูล)

Home	Quote	Fund	Stock	Portfolio	E-Alert
SET Index Trading Summary (TIPS:กดที่นี่ เพื่อแสดงข้อมูลย้อนหลัง)			SET INDEX Historical (*.csv)		
วันที่	กองทุนและ โบรกเกอร์	ต่างชาติ	รายย่อย	ข้อมูล EOD ปี 1970 ถึงปีที่แล้ว (.csv)	Download
2017-06-16	-59.45	+1002.36	-942.91	ข้อมูล EOD ตั้งแต่ปีถึงปัจจุบัน (.csv)	Download
2017-06-15	-543.77	-196.30	+740.07	ข้อมูล EOD วันล่าสุดวันเดียว (.csv)	Download
2017-06-14	+2318.62	-1212.70	-1105.92		
2017-06-13	+1687.66	-1528.93	-158.73		
2017-06-12	+766.41	-351.61	-414.80		

จากรูปภาพข้างบน ต้องดาวน์โหลด 2 อย่าง ได้แก่

- ข้อมูล EOD ปี 1970 ถึงปีที่แล้ว (.csv)
- ข้อมูล EOD ตั้งแต่ปีถึงปัจจุบัน (.csv)

จากนั้นให้แตกไฟล์ .csv ออกมาจาก 2 แหล่งข้อมูลดังกล่าว แล้วมากองรวมกันที่โฟลเดอร์

อย่างของผมแตกไฟล์ แล้วมารวมไว้ที่โฟลเดอร์ `set-archive_EOD_UPDATE`

ทั้งนี้ชื่อไฟล์ .csv จะเรียงตามวันที่ซื้อขาย (เริ่มตั้งแต่ซื้อขายวันแรก 30 เมษายน พ.ศ. 2518)

ซึ่งข้างในไฟล์จะเก็บข้อมูลหลักทรัพย์ทุกตัว ที่ซื้อขายในวันนั้นๆ

ตัวอย่างไฟล์ set-history_EOD_2017-05-23.csv

เป็นข้อมูลหลักทรัพย์ที่ซื้อขายเฉพาะในวันที่

2017-05-23

	A	B	C	D	E	F	G
1	<TICKER>	<DTYYYY>	<OPEN>	<HIGH>	<LOW>	<CLOSE>	<VOL>
2	IDIV	20170523	12.34	12.34	12.24	12.25	2300
3	BMSCITH	20170523	11.39	11.41	11.37	11.41	105400
4	CHINA	20170523	5.35	5.37	5.35	5.36	23000
5	EBANK	20170523	5.67	5.67	5.67	5.67	2600
6	ECOMM	20170523	7.43	7.43	7.43	7.43	1400
7	EFOOD	20170523	6.54	6.54	6.54	6.54	200
8	ENGY	20170523	5.18	5.18	5.17	5.17	4100
9	ENY	20170523	6.05	6.05	6.05	6.05	1800
10	GLD	20170523	1.86	1.87	1.86	1.86	145600
11	TDEX	20170523	9.76	9.76	9.72	9.75	302800
12	TGOLDETF	20170523	4.1	4.1	4.08	4.08	96900
13	TH100	20170523	4.54	4.6	4.54	4.6	222700
14	2S	20170523	5.65	5.95	5.6	5.8	2712500
15	A	20170523	6.55	6.55	6.55	6.55	100500
16	AAV	20170523	6.05	6.1	6.05	6.05	3953900
17	ABC	20170523	0.21	0.22	0.2	0.21	5430600
18	ABICO	20170523	10.8	11.5	10.8	11.4	3193600
19	ABPIF	20170523	8.35	8.4	8.35	8.35	97800
20	ACAP	20170523	20.7	20.9	20.5	20.8	4896700
21	ACC	20170523	0.58	0.58	0.57	0.57	217800
22	ADVANC	20170523	178	179	177.5	177.5	3993400
23	AEC	20170523	0.64	0.64	0.63	0.63	307200
24	AEONTS	20170523	106	109.5	106	108.5	25600

เนื่องจากข้อมูลที่ได้มา มันเอาไปใช้ลำบากนิดตดหนึ่ง

ผมเลยต้องจัดเตรียมข้อมูลใหม่ เพื่อให้ 1 ไฟล์เก็บเฉพาะข้อมูล 1 หลักทรัพย์ เช่น ไฟล์ PTT.csv ก็ให้มีเฉพาะข้อมูลหุ้น PTT ดังรูปข้างล่าง

	A	B	C	D	E	F	G
1	<TICKER>	Date	Open	High	Low	Close	Volume
2	PTT	20090318	149	149	147	148	2268505
3	PTT	20090319	148	153	148	149	6118953
4	PTT	20090320	152	153	150	151	4710828
5	PTT	20090323	153	161	152	160	17952782
6	PTT	20090324	162	163	158	161	11304913
7	PTT	20090325	160	161	158	160	6614355
8	PTT	20090326	160	164	159	161	11147069
9	PTT	20090327	162	163	159	160	6812569
10	PTT	20090330	157	157.5	151	152	8294191
11	PTT	20090331	153	154	152	152.5	4645097
12	PTT	20090401	153.5	153.5	151	151.5	5511284
13	PTT	20090402	153.5	159	153	158	9152612
14	PTT	20090403	159.5	162	158	162	9034324
15	PTT	20090407	161.5	164	159	160	7114034
16	PTT	20090408	158	160	158	160	4876618
17	PTT	20090409	160.5	164.5	160	161	6944874
18	PTT	20090410	163	166	160.5	165.5	7695763
19	PTT	20090416	164	166	160.5	163	20939408
20	PTT	20090417	165	170	163.5	169	17462714

21	PTT	20090420	169.5	182	168.5	178	19057308
22	PTT	20090421	174	177	173.5	176.5	10282582
23	PTT	20090422	178	179.5	171.5	173.5	10657065
24	PTT	20090423	174	177	173	176.5	5658580

แถมผมยังจะเปลี่ยนชื่อหัวคอลัมน์ เป็น Open, High, Low, Close, Volume (ไม่ใช่ “<OPEN>”, “<HIGH>”, “<LOW>”, “<CLOSE>”, “<VOL>”)

...ด้วยเหตุนี้ผมเลยต้องเขียน โค้ดขึ้นเพิ่มเติม

แต่ก่อนอื่นจะขอติดตั้งโมดูลเพิ่มเติม ได้แก่ tqdm ด้วยคำสั่งตามนี้

```
pip install tqdm
```

ถ้าถามว่าโมดูลนี้มีไว้ทำอะไร ก็ต้องบอกว่าเอาไว้แสดงสถานะแบบ progress bar บนคอมมานไลน์ ก็เวลาที่โค้ดมันทำงานจะนานสักนิดหนึ่ง เพราะข้อมูลมันเยอะ เลยต้องให้เห็นสถานะความคืบหน้าสักหน่อย ...คราวนี้จะมาดูโครงสร้างโปรเจกก่อนเขียนโค้ด

```
root_folder\
|-- datasets\
    |-- sec_csv\
    |-- set-archive_EOD_UPDATE\
    |-- siamchart.py
```

- **sec_csv** เป็นโฟลเดอร์ เอาไว้เก็บไฟล์ .csv (เอาท์พุต)
- **set-archive_EOD_UPDATE** เป็นโฟลเดอร์ เก็บไฟล์ .csv ที่ได้จาก <http://siamchart.com/stock/> (<http://siamchart.com/stock/>) (ที่เราดาวน์โหลดมา)
- **siamchart.py** คือโค้ดที่จะเขียนขึ้นมา

ตัวอย่างโค้ด

```
def getStockData(eodFiles, selectedSmbol = []):
    dict = {}          # empty dictionary
    for i, f in enumerate(eodFiles):# read all file
        df = pd.read_csv(f)
        total_row = len(df.index)
        all_data = df.values;

        #range(start, stop, step)
        for row in range(total_row-1, -1, -1):
            symbol = all_data[row][0]
            if len(selectedSmbol) != 0:
                if not symbol in selectedSmbol:
                    continue

            if symbol == "COM7":
                symbol = "COM7F"

            current_row = all_data[row][1]
            if symbol in dict:      # already have data
                dict[symbol] += current_row
            else: # no symbol data
                dict[symbol] = current_row

        if(i%500 == 0): # for debug
            print("Reading total file %d" % i)

    return dict
```



```

DIR_CURRENT = os.path.dirname(__file__)
DIR_SEC_CSV = "sec_csv"
# download: http://siamchart.com/stock/ (Must register to )
EOD_file = "set-archive_EOD_UPDATE"
def createSymbolCSV(start_idx, outputPath=DIR_SEC_CSV):
    eodFiles = getFileNameInDir(EOD_file)
    eodFiles = eodFiles[-1 * start_idx:] # select files

    outputPath = join(DIR_CURRENT,outputPath)
    clearDir(outputPath) # delete old files

    eodFiles = [ join(DIR_CURRENT,file) for file in eodFiles ]
    dataStock = getStockData(eodFiles)
    headers = getHeaderFile(eodFiles) # Read header of
    columnNames = { index:changeName(value) for index, value in headers.items() }

    # write data to csv files separate file name follow symbol
    count = 0
    for key, allRow in dataStock.items():
        df = pd.DataFrame(allRow)
        df.rename(columns=columnNames, inplace=True)
        fileName = "{}.csv".format(join(outputPath, key))
        df.to_csv(fileName, index = False) # write to csv

        if(count%3000 == 0): # for debug
            print("Writing total files: {}".format(count))

        count+=1;

```

วิธีใช้งานฟังก์ชัน

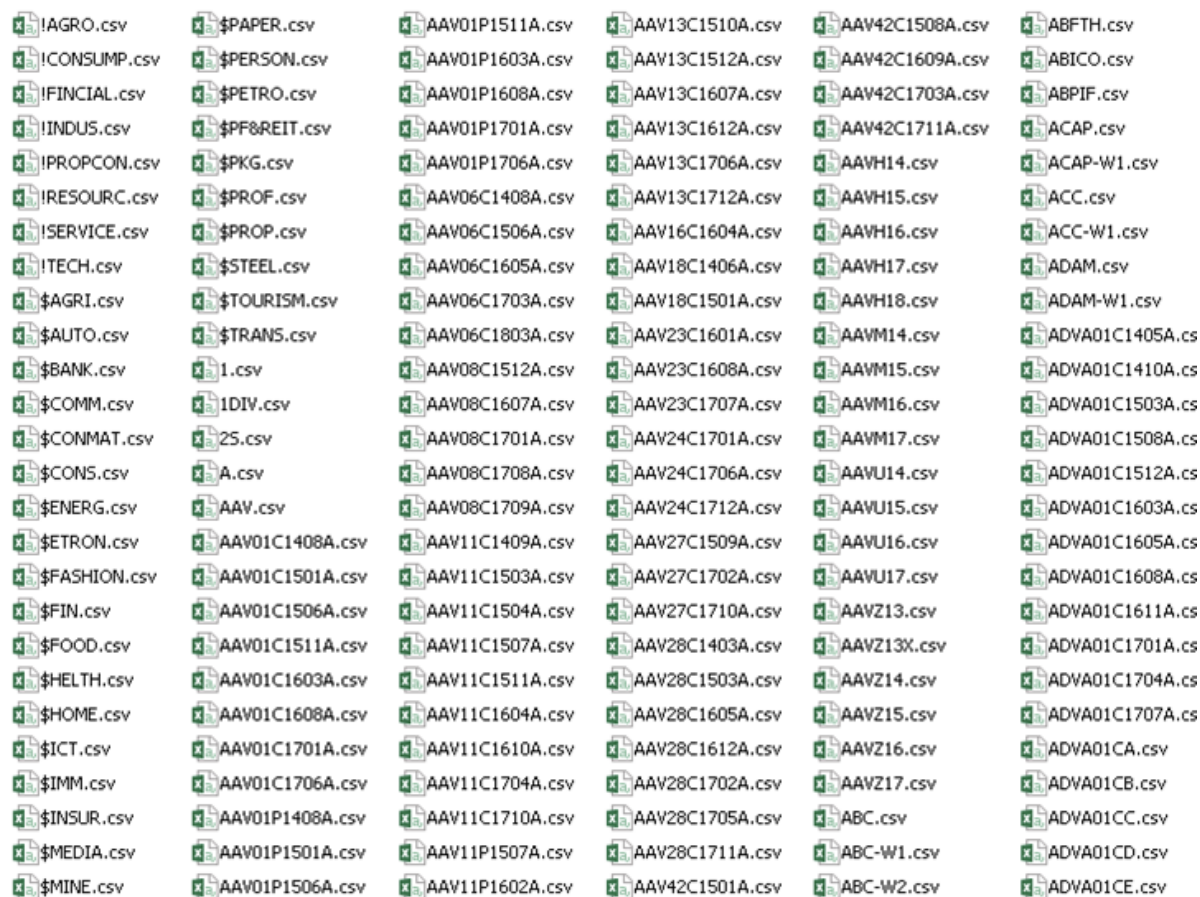
```
createSymbolCSV(2000)
```

โค้ดข้างบนจะสร้างไฟล์ .csv โดยใช้ข้อมูลของวันที่ปัจจุบัน นับย้อนหลังไป 2,000 วัน (ทำการ)
(ขอยกยผมไม่ได้นำโค้ดทั้งหมดมาแสดงให้ดู ถ้าจะดูโค้ดทั้งหมด ผมแปะไว้ในท้ายข้อที่ 4 นี้แล้วนะครับ)

เวลารันก็ใช้คำสั่ง

```
python siamchart.py
```

เมื่อโค้ดรันเสร็จ จะได้ไฟล์ .csv เป็นชื่อย่อของหลักทรัพย์นั้นๆ ดังรูปข้างล่าง



สาเหตุที่ผมแยก .csv ตามชื่อหลักทรัพย์แต่ละตัว เพราะจะทำให้การเขียนโค้ดสะดวกขึ้น ดังรายละเอียดในหัวข้อวิธีนำข้อมูลมาใช้

โค้ดวิธีที่ 4 เวอร์ชันเต็มทั้งหมด ก็ดาวน์โหลดได้ที่

- <https://github.com/adminho/trading-stock-thailand/blob/master/datasets/siamchart.py> (<https://github.com/adminho/trading-stock-thailand/blob/master/datasets/siamchart.py>)

วิธีที่ 5 Web scraping

ยกตัวอย่างหุ้น PTT เมื่อใช้ URL เป็น

<https://www.set.or.th/set/historicaltrading.do?symbol=PTT&page=2&language=en&country=US&type=trading> (<https://www.set.or.th/set/historicaltrading.do?symbol=BBL&page=2&language=en&country=US&type=trading>)

(ถ้าหุ้นตัวอื่นก็เปลี่ยนใน URL เช่น BBL ก็แก้ไขเป็น symbol=BBL)

ก็จะเห็นหน้าตาเว็บดังนี้ (ราคาหุ้น PTT)

The Stock Exchange of Thailand									
https://www.set.or.th/set/historicaltrading.do?symbol=PTT&page=2&language=en&country=US&type=trading									
Companies/Securities	Date	Open	High	Low	Close	Change	%Change	Total Volume (Shares)	Total Value (1000 Baht)
News	28/02/2017	400.00	400.00	397.00	397.00	-3.00	-0.75	3,006,690	1,197,195.24
Equities	27/02/2017	397.00	400.00	397.00	400.00	+1.00	+0.25	2,444,930	974,447.96
Companies/Securities in Focus	24/02/2017	397.00	400.00	396.00	399.00	+1.00	+0.25	3,360,486	1,337,601.43
Stock Calendar	23/02/2017	400.00	400.00	396.00	398.00	-4.00	-1.00	3,888,251	1,548,845.87
IPOs	22/02/2017	399.00	402.00	397.00	402.00	+5.00	+1.26	5,666,866	2,261,876.11
New Listed	21/02/2017	398.00	402.00	396.00	397.00	0.00	0.00	5,575,986	2,221,080.34
Companies/Securities List of	20/02/2017	397.00	398.00	395.00	397.00	+1.00	+0.25	2,187,436	867,599.92
Companies/Securities ISIN	17/02/2017	395.00	399.00	395.00	396.00	+2.00	+0.51	5,877,275	2,331,417.87
Bonds	16/02/2017	397.00	398.00	391.00	394.00	-3.00	-0.76	6,346,865	2,500,925.83
Opportunity Day & Company Highlights	15/02/2017	387.00	397.00	386.00	397.00	+11.00	+2.85	11,694,199	4,573,391.21
	14/02/2017	394.00	394.00	386.00	386.00	-10.00	-2.53	13,315,823	5,188,568.29
	10/02/2017	398.00	398.00	393.00	396.00	+2.00	+0.51	14,080,113	5,577,275.32
	09/02/2017	410.00	412.00	394.00	394.00	-14.00	-3.43	12,868,058	5,140,114.73

เมื่อมองเป็นภาษา HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif-->
4 <!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8" lang="en"> <![endif-->
5 <!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif-->
6 <!--[if gt IE 8]><!-- <html class="no-js" lang="en"> <!--<![endif-->
7 <head>
8 <meta charset="utf-8">
9 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1,IE=8">
10 <meta name="description" content="The Stock Exchange of Thailand: Your Investment Resource for Thailand's Capital Market">
11 <meta name="keywords" content="SET, mai, thai stocks, Thailand, stock, stock exchange, Thai capital market, equity, bond, derivatives, stock market, quotes, financial, internet trading, listed companies, IPO, regulations, broker, market data, investment information, news, investor education">
12 <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=yes">
13
14
15
16
17 <title>The Stock Exchange of Thailand - Companies/Securities in Focus</title>
18
19 <link rel="shortcut icon" href="/images/favicon.ico" type="image/vnd.microsoft.icon" />
20 <link rel="stylesheet" href="/styles/bootstrap.min.css">
21 <link rel="stylesheet" href="/styles/bootstrap-theme.min.css">
22 <link rel="stylesheet" href="/styles/font-awesome.min.css">
23 <link rel="stylesheet" href="/styles/bootstrap-submenu.min.css">
24 <link rel="stylesheet" href="/styles/main.css">
25 <link rel="stylesheet" href="/styles/main-en.css">
26 <link rel="stylesheet" href="/styles/setstyle.css">
27 <link rel="stylesheet" href="/styles/main-responsive.css">
28 <link rel="stylesheet" href="/styles/tablet.css">
29 <link rel="stylesheet" href="/scripts/fullcalendar-2.1.1/fullcalendar.min.css">
30 <!--[if lt IE 10]><link rel="stylesheet" href="/styles/ie.css"> <![endif-->
31 <script src="/scripts/vendor/jquery-1.11.2.min.js"></script>
32 <!-- Start Other CSS -->
33 <!-- Start Chart CSS -->
34 <link rel="stylesheet" href="/styles/chart.css">
35 <!-- End Chart CSS -->
36 <!-- End Other CSS -->
37 <!--[if lt IE 9]>
38 <script src="/scripts/vendor/html5shiv.min.js"></script>
```

Web scraping (https://en.wikipedia.org/wiki/Web_scraping) คือการดึงข้อมูลจากหน้าเว็บ โดยในทางโปรแกรมมิ่ง เนื้อหา HTML เราจะมองเป็นโครงสร้าง DOM (https://www.w3schools.com/js/js_htmlDOM.asp) แล้วก็สามารถเข้าถึงทีละ element ได้

ด้วยเหตุนี้เราจึงสามารถเขียนโค้ดดึงข้อมูลราคาหุ้น PTT ได้ในภาษา Python ดังตัวอย่าง

```
def getTableData(symbol, page=1):
    if page > 3:
        page = 3 # limit at 3

    url_string = "https://www.set.or.th/set/historicaltrading.do?symbol=" + symbol
    url_string += '&page={0}&language=en&country=US&type=trading'.format(page)
    page = urllib.request.urlopen(url_string).read()
    soup = BeautifulSoup(page, 'lxml')
    table_element = soup.find('table', class_='table table-hover table-striped')
    return table_element, url_string
```

ลองทดสอบรันฟังก์ชัน

```
table_element, url_string = getTableData("PTT")
tr_list = table_element.findAll('tr')
print(tr_list[0:2])
```

ได้ผลลัพธ์ (โชว์ข้อมูลแค่ส่วน header ของตาราง กับ ข้อมูลหุ้น PTT แถวแรก ถ้าโชว์หมดไมไหว มันเยอะไป)

```
[<tr align="center" class="bggrey1">
  <th height="35" width="10%"><strong>Date</strong></th>
  <th width="10%"><strong>Open</strong></th>
  <th width="10%"><strong>High</strong></th>
  <th width="10%"><strong>Low</strong></th>
  <th width="10%"><strong>Close</strong></th>
  <th width="10%"><strong>Change</strong></th>
  <th width="10%"><strong>%Change</strong></th>
  <th width="13%"><strong>Total Volume<br/>(Shares)</strong></th>
  <th width="17%"><strong>Total Value<br/>('000 Baht)</strong></th>
</tr>, <tr align="right">
  <td align="center">16/06/2017</td>
  <td>380.00</td>
  <td>382.00</td>
  <td>377.00</td>
  <td>379.00</td>
  <td><font color="#FF0000"> -1.00</font></td>
  <td><font color="#FF0000"> -0.26</font></td>
  <td>4,051,989</td>
  <td>1,537,504.20</td>
</tr>]
```

ดูโค้ดวิธีที่ 5 ฉบับเต็มได้ที่

- https://github.com/adminho/trading-stock-thailand/blob/master/datasets/web_set_price.py (https://github.com/adminho/trading-stock-thailand/blob/master/datasets/web_set_price.py)
- ในโค้ดจะมีการเซฟข้อมูลเก็บไว้เป็นไฟล์ .csv ที่โฟลเดอร์ DIR_SEC_CSV = "sec_set_price"

แต่มีข้อควรระวัง วิธีที่ 5 ด้วย Web scraping เหมือนดาบสองคม ถ้าเรียกใช้ติดต่อกันถี่ๆ ก็เหมือนเราส่ง request ไปโจมตี Serverเดี๋ยวเจอเพ่งเล็งจากตลาดหลักทรัพย์ได้ว่า เป็นการโจมตีแบบ DOS (<https://th.wikipedia.org/wiki/การโจมตีโดยปฏิเสธการให้บริการ>) บทความนี้ไม่แนะนำ เอาไปใช้ดูหุ้นแบบ real time นะครับ

วิธีที่ 6 สุดท้าย (ไม่รู้จะนับรวมดีไหม)

ยังมีวิธีเรียกดูข้อมูลด้วยโมดูล yahoo_finance
ติดตั้งโมดูลด้วยคำสั่ง

```
pip install yahoo_finance
```

ตัวอย่างโค้ด ดูหุ้นฝรั่ง YHOO และค่าเงิน EURPLN

```
from yahoo_finance import Share
yahoo = Share("YHOO")
print("\n++++YHOO stock++++")
print (yahoo.get_open())
print (yahoo.get_price())
print (yahoo.get_trade_datetime())

from yahoo_finance import Currency
eur_pln = Currency('EURPLN')
print("\n++++Currency of EURPLN++++")
print (eur_pln.get_bid())
print (eur_pln.get_ask())
print (eur_pln.get_rate())
print (eur_pln.get_trade_datetime())
```

ได้ผลลัพธ์

```
++++YHOO stock++++
None
52.58
2017-06-16 20:00:00 UTC+0000

++++Currency of EURPLN++++
4.2100
4.2110
4.2100
2017-06-19 16:10:00 UTC+0000
```

แต่อนิจจิง วัฏสังขาร ...ผมไม่รู้วิธีเรียกดูข้อมูลหุ้นไทยอะนะ ใครทำเป็นบอกที

ถ้าเียงลองอ่านเพิ่มเติมได้ที่

<https://pypi.python.org/pypi/yahoo-finance>

(<https://pypi.python.org/pypi/yahoo-finance>)

ดูโค้ดตัวอย่างได้ที่ (โค้ดจะกองรวมกันอยู่กับวิธีในข้อ 1-3)

- <https://github.com/adminho/trading-stock-thailand/blob/master/datasets/miscellaneous.py> (<https://github.com/adminho/trading-stock-thailand/blob/master/datasets/miscellaneous.py>)

วิธีนำข้อมูลมาใช้

ตอนแรก ถ้าจบแค่การดึงข้อมูลหุ้นไทยออกมา คงกะไรอยู่ ผมเลยต้องขอยอดด้วยการนำเอาข้อมูลมาใช้ต่อ เป็นพื้นฐานนะ...ซึ่งจะเอาข้อมูลเฉพาะที่ได้จากวิธีที่ 4 รวมทั้งโค้ดด้วย เอามาใช้งานต่อ (ขอตกลงตามนี้ก่อนนะ)

ตัวอย่างอ่านข้อมูลเฉพาะหุ้น PTT มาแสดงผล

```
def load_OHLCV(symbol, dates,
               column_names=['Open', 'High', 'Low', 'Close', 'Volume'],
               base_dir=DIR_SEC_CSV):
    """Read securities data for given symbols from
    df_main = pd.DataFrame(index=dates) # en

    if 'Date' not in column_names:
        column_names = np.append

    base_dir = join(DIR_CURRENT, base_dir)
    csv_file = os.path.join(base_dir, "{}.csv".format(symbol))
    df_csv = pd.read_csv(csv_file, index_col='Date',
                        parse_dates=True, usecols=column_names)
    df_main = df_main.join(df_csv)
    df_main = df_main.dropna(0)

    return df_main
```

เรียกฟังก์ชันให้ทำงาน โดยข้อมูลเริ่มต้นคือวันที่ '2017-03-01' ถึงวันที่ปัจจุบัน (ตอนที่เขียนบทความคือวันที่ 2017-06-16)

```
startDate = '2017-03-01'
endDate = strftime("%Y-%m-%d", gmtime())
dates = pd.date_range(startDate, endDate)
df = load_OHLCV("PTT", dates)
print(df.tail())
```

ได้ผลลัพธ์

	Open	High	Low	Close	Volume
2017-06-12	384.0	384.0	380.0	381.0	2245000.0
2017-06-13	382.0	386.0	379.0	385.0	6457400.0
2017-06-14	385.0	386.0	383.0	384.0	2672300.0
2017-06-15	383.0	384.0	379.0	380.0	5239600.0
2017-06-16	380.0	382.0	377.0	379.0	4050700.0

ผลลัพธ์จะแสดงค่าข้อมูลหุ้น PPT ได้แก่ ราคาเปิด (Open) ราคาสูงสุด (High) ราคาต่ำสุด (Low) ราคาปิด (Close) และปริมาณการซื้อขาย (Volume) ของ 5 วันล่าสุด

ตัวอย่างพล็อตกราฟแท่งเทียนของหุ้น PTT

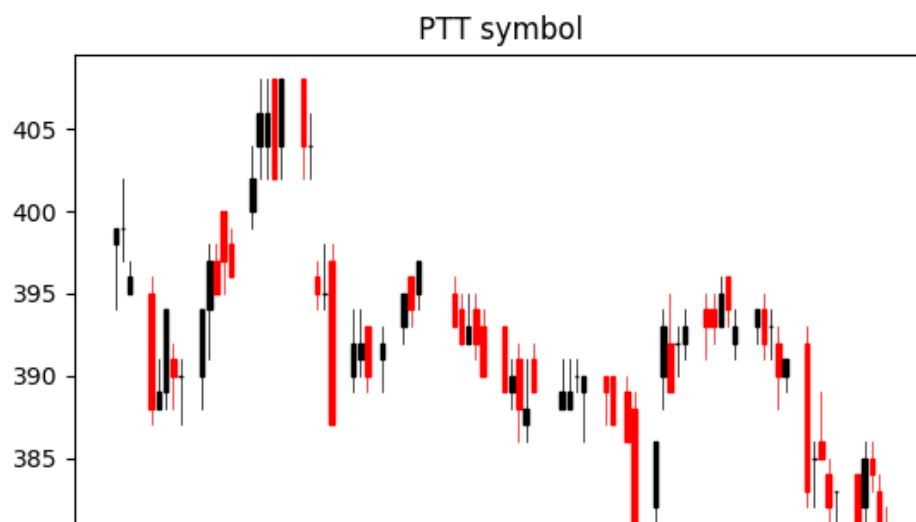

```
# Borrowed code from : http://matplotlib.org/examples/pylab\_examples/finance\_tutorial.html
def plotCandlestick(symbol, dates, title="Selected data"):
    quotes = loadStockQuotes(symbol, dates)
    mondays = WeekdayLocator(MONDAY)          # major ticks on Mondays
    alldays = DayLocator()                      # minor ticks on all days
    weekFormatter = DateFormatter('%b %d')      # e.g., Jan 12
    dayFormatter = DateFormatter('%d')         # e.g., 12

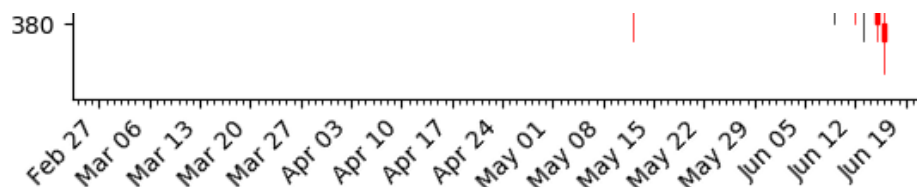
    fig, ax = plt.subplots()
    fig.subplots_adjust(bottom=0.2)
    ax.xaxis.set_major_locator(mondays)
    ax.xaxis.set_minor_locator(alldays)
    ax.xaxis.set_major_formatter(weekFormatter)
    #ax.xaxis.set_minor_formatter(dayFormatter)
    #plot_day_summary(ax, quotes, ticksize=3)
    candlestick_ohlc(ax, quotes, width=0.6)
    ax.xaxis_date()
    ax.autoscale_view()
    ax.set_title(title)
    plt.setp(plt.gca().get_xticklabels(), rotation=45, horizontalalignment='right')
    plt.show()
```

เรียกใช้ฟังก์ชัน

```
plotCandlestick("PTT", dates, title = "PTT symbol")
```

ได้ผลลัพธ์





ตัวอย่าง หุ้น PTT, AOT, SCC, CPALL จะเอาเฉพาะราคาปิด (Close) มาแสดงพร้อมกันทั้ง 5 ตัว (รวม SET ด้วย)

```
def loadManySymbols(symbols, dates, column_name, base_dir):
    """Read securities data for given symbols from CSV file"""
    df = pd.DataFrame(index=dates) # empty data frame
    if 'SET' not in symbols: # add SET for reference, if a
        symbols = np.append(['SET'],symbols)

    base_dir = join(DIR_CURRENT,base_dir)
    for symbol in symbols:
        # read CSV file path given ticker symbol
        csv_file = os.path.join(base_dir, symbol)
        df_temp = pd.read_csv(csv_file, index_col=0,
                               parse_dates=True, usecols=[column_name])

        df_temp = df_temp.rename(columns={column_name: symbol})
        df = df.join(df_temp) # left join by default
        if symbol == 'SET': # drop dates SET does not have
            df = df.dropna(subset=[symbol])

    return df
```

```
def loadPriceData(symbols, dates, base_dir=DIR_SEC_CSV):
    return loadManySymbols(symbols, dates, 'Close', base_dir)
```

เรียกใช้ฟังก์ชัน

```
symbols = ["PTT", "AOT", "SCC", "CPALL"]
df = loadPriceData(symbols, dates)
print(df.tail())
```

ได้ผลลัพธ์ (เอาข้อมูล 5 วันล่าสุด มาแสดง)

	SET	PTT	AOT	SCC	CPALL
2017-06-12	1563.81	381.0	44.00	520.0	61.75
2017-06-13	1572.36	385.0	44.25	518.0	62.00
2017-06-14	1577.00	384.0	44.25	520.0	62.00
2017-06-15	1573.53	380.0	44.50	516.0	61.75
2017-06-16	1576.58	379.0	45.25	518.0	61.75

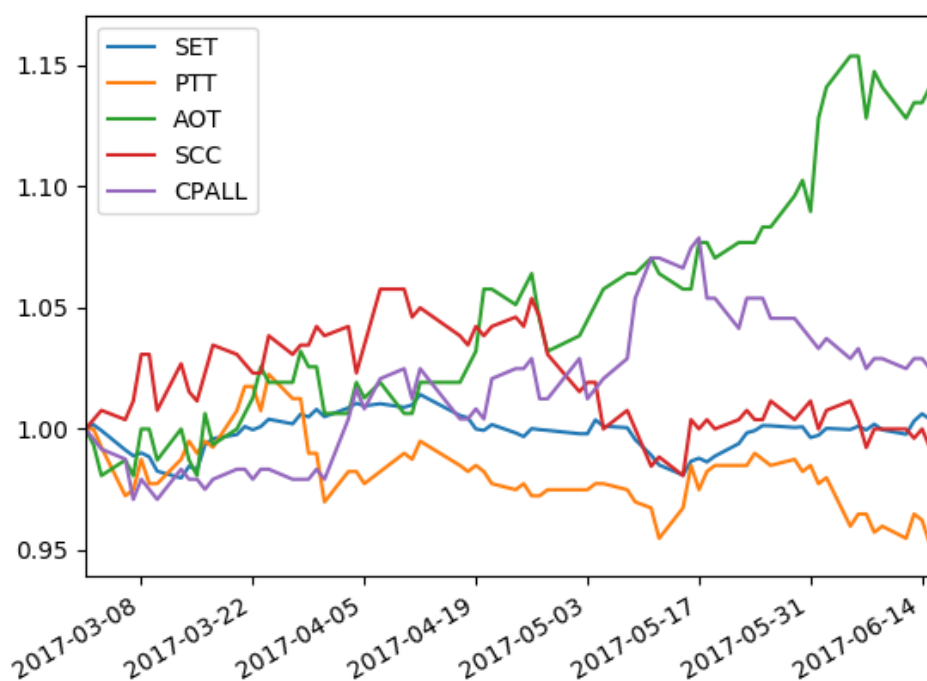
หมายเหตุ ฟังก์ชัน `loadManySymbols()` จะแอบโหลดข้อมูล SET มาแปะไว้เป็นคอลัมน์แรกเสมอ

หรือจะพล็อตกราฟรวม

```
df = df/df.iloc[0,:]
df.plot()
plt.show()
```

ในโค้ดบรรทัดที่เขียน `df = df/df.iloc[0,:]` จะเป็นการทำ normalized อย่างง่ายๆ โดยหุ้นแต่ละตัวจะเอาข้อมูล ณ วันที่เริ่มต้นมาหารวันอื่นที่เหลือทั้งหมด (บทความนี้ วันที่เริ่มต้นคือ '2017-03-01')

ที่ทำแบบนี้จะได้เห็นภาพหุ้นแต่ละตัวในกราฟเดียวกันง่ายขึ้น (มีสเกลเดียวกัน) ดังรูปข้างล่าง



สำหรับโค้ดส่วนนี้ เวอร์ชันเต็มทั้งหมด ก็ดาวน์โหลดที่

- https://github.com/adminho/trading-stock-thailand/blob/master/dataset_siamchart.py (<https://github.com/adminho/trading-stock-thailand/blob/master/datasets/siamchart.py>)

หมายเหตุ เนื่องจากการดาวน์โหลดไฟล์ EOD จากเว็บ <http://www.siamchart.com/> (<http://www.siamchart.com/>) อาจดูยุ่งยากไป

จึงอาจใช้วิธีที่ 3 (โมดูล pandas-datareader) ซึ่งจะดาวน์โหลดข้อมูลแบบออนไลน์ได้โดยตรง (แต่ยังมีปัญหาตรงที่ดาวน์โหลดข้อมูล SET ไม่ได้)

และผมก็ทำตัวอย่างซอร์สโค้ดไว้ที่

- https://github.com/adminho/trading-stock-thailand/blob/master/datasets/pandas_reader.py (https://github.com/adminho/trading-stock-thailand/blob/master/datasets/pandas_reader.py)

โดยมีชื่อฟังก์ชันและใช้งานได้เหมือนกันดังนี้ (พารามิเตอร์ต่างกันนิดหน่อย)

- load_OHLCV
- plotCandlestick
- loadPriceData (ไม่ได้แปะข้อมูล SET ที่คอลัมน์แรก)

คำนวณ Indicator อย่างง่าย

จะกำหนดโครงสร้างโปรเจคใหม่

```
root_folder\
|-- datasets\
    |-- sec_csv\
    |-- set-archive_EOD_UPDATE\
    |-- siamchart.py
|-- indicator_example.py
```

ไฟล์ `indicator_example.py` เอาไว้ใช้คำนวณ Indicator อย่างง่าย โดยข้างในจะอิมพอร์ต `siamchart.py` เข้ามาอีกที ดังนี้

```
import datasets.siamchart as ds
```

ตัวอย่างคำนวณค่า EMA ของหุ้น PTT, AOT, SCC, CPALL (รวม SET ด้วย)

EMA (Exponential Moving Average) หรือก็คือ เส้นค่าเฉลี่ยเคลื่อนที่แบบเอกซ์โพเนนเชียล สำหรับนักลงทุนสายเทคนิคคงรู้จักกันดีอยู่แล้ว ใครไม่ทราบลองอ่านดูที่นี่แล้วกันครับ (http://www.aommoney.com/daddytrader/ร่ายด้วยหุ้น-4-ema#gs._4KZuBI)

ส่วนที่มาสูตรคงไม่ลงลึกนะครับ ลองหาอ่านดูที่นี่ (http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_averages)

วิธีคำนวณ สามารถเขียนโค้ดได้ดังนี้

```
def ema(df, periods=12):
    return df.ewm(span=periods, adjust=True, min_periods=0, ignore_na=True).ewm_mean()
```

ลองทดสอบฟังก์ชัน ด้วยการหา EMA (15 วัน) ของหุ้นทั้ง 5 ตัว (รวม SET ด้วย)

```
startDate = '2017-03-01'
endDate = strftime("%Y-%m-%d", gmtime())
dates = pd.date_range(startDate, endDate)
symbols = ["PTT", "AOT", "SCC", "CPALL"]
df = ds.loadPriceData(symbols, dates)
ema15 = ema(df, 15)
print(ema15.tail())
```

ได้ผลลัพธ์ (แสดงเฉพาะข้อมูลล่าสุด 5 ตัว)

	SET	PTT	AOT	SCC	CPALL
2017-06-12	1565.149192	386.862403	43.625811	521.696472	62.367670
2017-06-13	1566.050622	386.629583	43.703841	521.234373	62.321708
2017-06-14	1567.419399	386.300860	43.772116	521.080064	62.281491
2017-06-15	1568.183275	385.513200	43.863108	520.445014	62.215050
2017-06-16	1569.232927	384.699002	44.036479	520.139369	62.156916

ตัวอย่างคำนวณค่า **MACD** ของหุ้น PTT, AOT, SCC, CPALL (รวม SET ด้วย)

ตัวเลข **MACD** (Moving Average Convergence Divergence) อ่านว่า “Mac-Dee” หรือจะได้เรียกว่า “M-A-C-D” ก็ได้ ซึ่งนักลงทุนสายเทคนิคคงรู้จักกันดีอยู่แล้ว ใครไม่ทราบลองอ่านดูที่นี่แล้วกันครับ (<http://www.aommoney.com/daddytrader/บทความทางเทคนิค/ตัวช่วย-ร่ายด้วยหุ้นเทค>)

-2#gs.nDsbzQM)

ส่วนที่มาสูตรคงไม่ลงลึกนะครับ ลองหาอ่านดูได้ที่นี้ (http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_average_convergence_divergence_macd)

วิธีคำนวณ สามารถเขียนโค้ดได้ดังนี้

```
def average_convergence(df, period_low=26, period_fast=12):
    """
    compute the MACD (Moving Average Convergence/Divergence)
    using a fast and slow exponential moving average'
    """
    emaslow = ema(df, period_low)
    emafast = ema(df, period_fast)
    return (emaslow, emafast, emafast - emaslow)
```

ลองทดสอบฟังก์ชัน ด้วยการหา MACD ของหุ้นทั้ง 5 ตัว (รวม SET ด้วย) โดยใช้ตัวแปร df จากตัวอย่างก่อน

```
_, _, macd = average_convergence(df)
print(macd.tail())
```

ได้ผลลัพธ์ (แสดงเฉพาะข้อมูลล่าสุด 5 ตัว)

	SET	PTT	AOT	SCC	CPALL
2017-06-12	1.259309	-2.217163	0.895479	-2.101097	-0.123858
2017-06-13	1.701377	-2.143972	0.860915	-2.207254	-0.137174
2017-06-14	2.396993	-2.141642	0.824020	-2.106336	-0.146042
2017-06-15	2.638832	-2.433321	0.805589	-2.321180	-0.171195
2017-06-16	3.040693	-2.713592	0.841591	-2.304025	-0.188948

ตัวอย่างพล็อตกราฟของหลักทรัพย์ทั้งหมด พร้อมค่า EMA 15 วัน, EMA 45 วัน และ EMA 100 วัน

ตัวอย่างโค้ดฟังก์ชันเอาไว้อพล็อตกราฟ

```
def plot_graph(values):
    style = ['b-', 'r-', 'g-', 'k-', 'y-']
    column_list = values[0].columns
    num_stock = len(column_list)
    date = values[0].index

    if num_stock > 6:
        num_stock = 6 # Fix si

    for pos in range(0, num_stock):
        column = column_list[pos]
        plt.subplot(2, 3, pos+1)

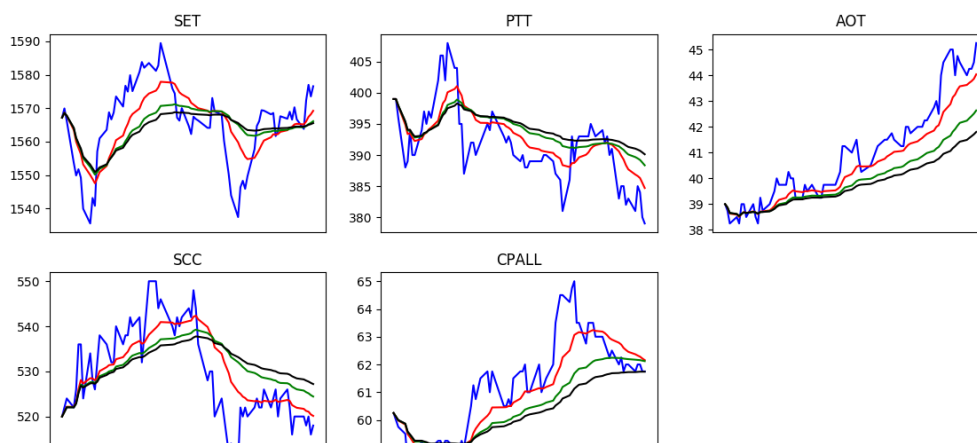
        for i in range(0, len(

plt.show()
```

นำตัวแปร df และ ema15 จากหัวข้อก่อนหน้านี้มาเรียกใช้ พร้อมทั้งเพิ่มตัวแปร ema45 และ ema100 เอาไว้เก็บค่า EMA 45 วัน และ EMA 100 ตามลำดับ

```
ema45 = ema(df, 45)
ema100 = ema(df, 100)
values = [df, ema15, ema45, ema100]
plot_graph(values)
```

ได้ผลลัพธ์



510 |  

ในกราฟแต่ละรูป เส้นสีน้ำเงินคือราคาของหลักทรัพย์แต่ละตัว

ส่วนเส้นสีแดง เขียว และดำ ในแต่ละกราฟ มันคือเส้น EMA 15 วัน , EMA 45 วัน และ 100 วันตามลำดับ

สำหรับโค้ดส่วนนี้ เวอร์ชันเต็มทั้งหมด ก็ดาวน์โหลดที่

- https://github.com/adminho/trading-stock-thailand/blob/master/indicator_example.py (https://github.com/adminho/trading-stock-thailand/blob/master/indicator_example.py)

่ออีกอย่าง ผมก็ทำตัวอย่างการคำนวณ Indicator ตัวอื่นไว้ด้วย โค้ดดรกๆ หน่อย ไม่เป็นระเบียบ และยังไม่เสร็จดีด้วยครับ ตัวอย่าง เช่น

ROC, Bollinger Band (BBANDS), daily returns, SMA, EMA, MACD, RSI, Sharpe ratio, True Range (TR), ATR, Beta, OBV และอื่นๆ

เพื่อใครสนใจโค้ดก็ได้ที่นี่

- <https://github.com/adminho/trading-stock-thailand/blob/master/indicator.py> (<https://github.com/adminho/trading-stock-thailand/blob/master/indicator.py>)

หวังว่าบทความนี้จะมีประโยชน์ต่อทุกคนนะครับ และถ้าเขียนผิดพลาดตรงไหนทักได้นะครับ

เพิ่มเติมให้ สำหรับใครที่อยากเอาโค้ดไปพัฒนาต่อ

สำหรับการนำ Python ไปใช้จัดการข้อมูล อย่างในบทความนี้คือหุ้นเป็นหลัก
ก็จะใช้ไลบรารี 4 อย่าง ที่เป็นพระเอกดังนี้

1 Panda

จะใช้สร้างโครงสร้างข้อมูลแบบตาราง จะช่วยให้เราสามารถประมวลผลทั้งตารางได้สะดวกมากขึ้น รวมทั้ง export ออกมาเป็น excel, csv, หรือ text ไฟล์ก็สะดวก ดูสรุปการใช้ไลบรารีนี้ได้ดังภาพข้างล่าง

Python For Data Science Cheat Sheet

Pandas

Learn Python for Data Science interactively at www.datacamp.com

Reshaping Data

Pivot

```
df = df.pivot(index='Date',
              columns='Type',
              values='Volume')
df
```

Spread rows into columns

Date	Type	Volume
2016-01-01	Buy	1000
2016-01-01	Sell	500
2016-01-02	Buy	2000
2016-01-02	Sell	1000
2016-01-03	Buy	3000
2016-01-03	Sell	1500

Pivot Table

```
df = pd.pivot_table(df,
                    index='Date',
                    columns='Type',
                    values='Volume',
                    aggfunc='sum')
df
```

Spread rows into columns

Date	Type	Volume
2016-01-01	Buy	1000
2016-01-01	Sell	500
2016-01-02	Buy	2000
2016-01-02	Sell	1000
2016-01-03	Buy	3000
2016-01-03	Sell	1500

Advanced Indexing

Selecting

```
df[df['Volume'] > 1000]
df[df['Volume'] > 1000].sort_values('Volume')
df[df['Volume'] > 1000].sort_index()
```

Indexing With Iloc

```
df.iloc[0:2, 1:2]
df.ix[0:2, 1:2]
```

Setting/Resetting Index

```
df.set_index('Date')
df.reset_index()
```

Reindexing

```
df.reindex(index=df.index.union(['2016-01-04']),
           fill_value=0)
```

Also see NumPy Arrays

Select with with/without values

```
df[df['Volume'] > 1000]
df[df['Volume'] > 1000].sort_values('Volume')
```

Select with with/without NaN

```
df[df['Volume'].notnull()]
df[df['Volume'].isnull()]
```

Subset the data

```
df[df['Volume'] > 1000]
df[df['Volume'] > 1000].sort_values('Volume')
```

Combining Data

Merge

```
df1 = pd.DataFrame({'key': 'key', 'val': 1})
df2 = pd.DataFrame({'key': 'key', 'val': 2})
df1.merge(df2, on='key')
```

key	val
key	1
key	2

Melt

```
df = pd.DataFrame({'Date': '2016-01-01',
                  'Type': 'Buy',
                  'Volume': 1000})
df
```

Gather columns into rows

Date	Type	Volume
2016-01-01	Buy	1000

Duplicate Data

```
df = pd.DataFrame({'Date': '2016-01-01',
                  'Type': 'Buy',
                  'Volume': 1000})
df
```

Return unique values
Check duplicates
Drop duplicates

Date	Type	Volume
2016-01-01	Buy	1000
2016-01-01	Buy	1000

Grouping Data

Aggregation

```
df.groupby('Date').mean()
```

Transformation

```
df.groupby('Date').transform(lambda x: x * 2)
```

Missing Data

```
df.dropna()
```

Drop NaN values
Fill NaN values with a predetermined value
Replace values with others

Date	Type	Volume
2016-01-01	Buy	1000
2016-01-01	Buy	1000

Dates

```
df['Date'] = pd.to_datetime(df['Date'])
df
```

Horizontal/Vertical

```
df.melt()
```

Date	Type	Volume
2016-01-01	Buy	1000

Visualization

```
df.plot()
```

Also see Matplotlib

Date	Type	Volume
2016-01-01	Buy	1000

Check Out Pandas Cheat Sheet

(https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Pandas_Cheat_Sheet_2.pdf)

ที่มา https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Pandas_Cheat_Sheet_2.pdf
(https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Pandas_Cheat_Sheet_2.pdf)

Data Wrangling
with pandas
Cheat Sheet
<http://pandas.pydata.org>

Tidy Data – A foundation for wrangling in pandas

In a tidy data set:

- Each variable is saved in its own column
- Each observation is saved in its own row

Tidy data complements pandas's **vectorized operations**. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.

$M * A$

Syntax – Creating DataFrames

```
df = pd.DataFrame({
    "a": [4, 5, 6],
    "b": [7, 8, 9],
    "c": [10, 11, 12]},
    index=[1, 2, 3])
```

Specify values for each column.

```
df = pd.DataFrame([
    [4, 7, 10],
    [5, 8, 11],
    [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
```

Specify values for each row.

```
df = pd.DataFrame({
    "a": [4, 5, 6],
    "b": [7, 8, 9],
    "c": [10, 11, 12]},
    index=pd.MultiIndex.from_tuples(
        [('d', 1), ('d', 2), ('e', 2)],
        names=['n', 'v']))
```

Create DataFrame with a MultiIndex

Reshaping Data – Change the layout of a data set

pd.melt(df)
Gather columns into rows.

df.pivot(columns='var', values='val')
Spread rows into columns.

pd.concat([df1, df2])
Append rows of DataFrames

pd.concat([df1, df2], axis=1)
Append columns of DataFrames

df.sort_values('mpg')
Order rows by values of a column (low to high).

df.sort_values('mpg', ascending=False)
Order rows by values of a column (high to low).

df.rename(columns = {'y': 'year'})
Rename the columns of a DataFrame

df.sort_index()
Sort the index of a DataFrame

df.reset_index()
Reset index of DataFrame to row numbers, moving index to columns.

df.drop(['Length', 'Height'], axis=1)
Drop columns from DataFrame

Subset Observations (Rows)

```
df[df.Length > 7]
```

Extract rows that meet logical criteria.

```
df.drop_duplicates()
```

Remove duplicate rows (only considers columns).

```
df.head(n)
```

Select first n rows.

```
df.tail(n)
```

Select last n rows.

```
df.sample(frac=0.5)
```

Randomly select fraction of rows.

```
df.sample(n=10)
```

Randomly select n rows.

```
df.iloc[10:20]
```

Select rows by position.

```
df.nlargest(n, 'value')
```

Select and order top n entries.

```
df.nsmallest(n, 'value')
```

Select and order bottom n entries.

Subset Variables (Columns)

```
df[['width', 'length', 'species']]
```

Select multiple columns with specific names.

```
df['width'] or df.width
```

Select single column with specific name.

```
df.filter(regex='regex')
```

Select columns whose names matches regular expression regex.

regex (Regular Expressions)	Examples
^.	Matches strings containing a period '.'
Length\$	Matches strings ending with word 'Length'
^Sepal	Matches strings beginning with the word 'Sepal'
^[1-5]\$	Matches strings beginning with '1' and ending with 1,2,3,4,5
^(?!(Species\$)).*	Matches strings except the string 'Species'

```
df.loc[:, 'x2': 'x4']
```

Select all columns between x2 and x4 (inclusive).

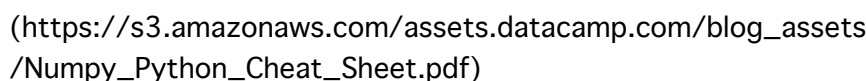
```
df.iloc[:, 1, 2, 5]
```

Select columns in positions 1, 2 and 5 (first column is 0).

(https://github.com/pandas-dev/pandas/blob/master/doc/cheatsheet/Pandas_Cheat_Sheet.pdf)

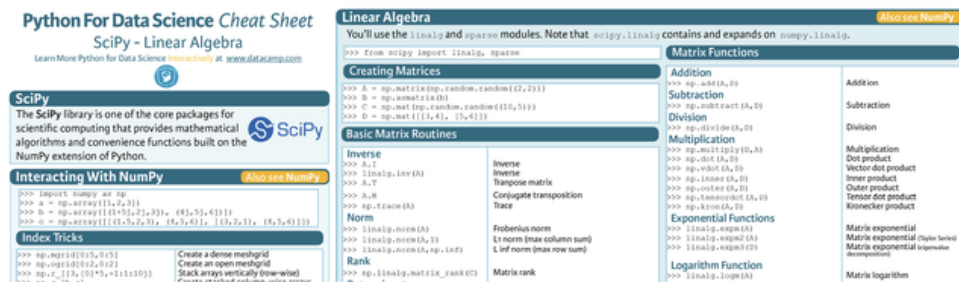
2 Numpy

เอาไว้คำนวณด้านตัวเลข จัดการพวกแอเรียร์ ทำได้อย่างสะดวก ดสรุปการใช้โลบรารีนี้ได้ดังภาพข้างล่าง



3 Scipy

เอาไว้คำนวณด้านเลข วิทยาศาสตร์ วิศวกรรม ดสรุปการใช้โลบริรี่นี้ได้ดังภาพข้างล่าง



Check Out SciPy Cheat Sheet

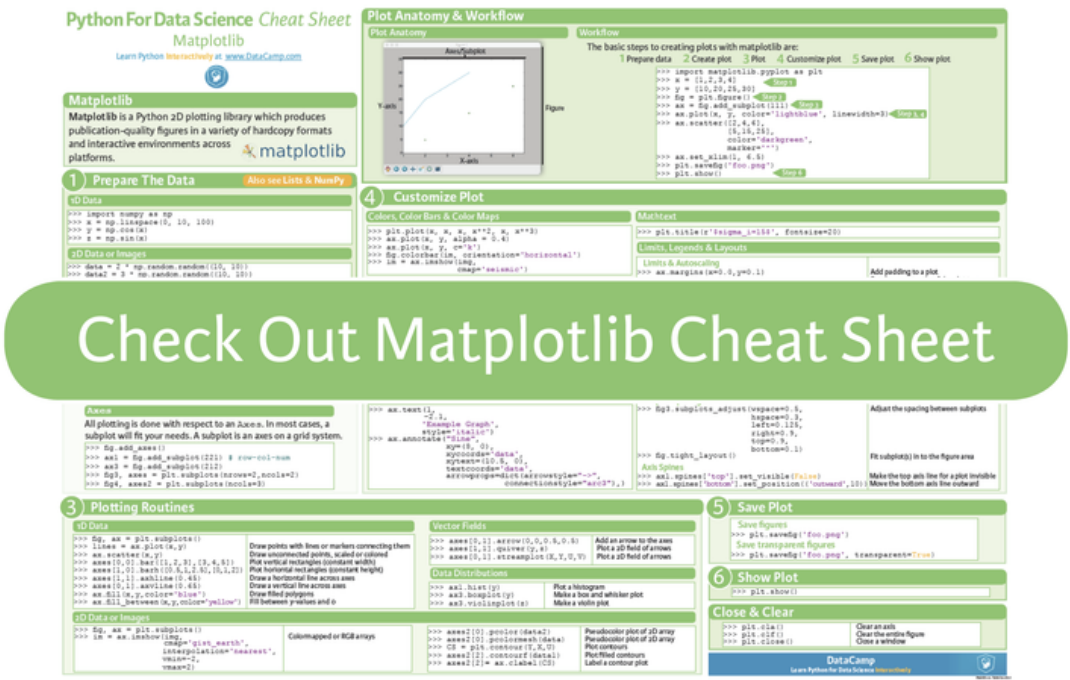


(https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_SciPy_Cheat_Sheet_Linear_Algebra.pdf)

ที่มา https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_SciPy_Cheat_Sheet_Linear_Algebra.pdf (https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_SciPy_Cheat_Sheet_Linear_Algebra.pdf)

4 Matplotlib

เอาไว้อัปโหลดกราฟ ดูสรุปการใช้ไลบรารีนี้ได้ดังภาพข้างล่าง



(https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Matplotlib_Cheat_Sheet.pdf)

ที่มา https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Matplotlib_Cheat_Sheet.pdf
(https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Matplotlib_Cheat_Sheet.pdf)

เขียนโดย

แอดมิน โฮ

โอน้อยออก

(<https://www.google.com/+PatanaPatanasongsivilai?rel=author>)

 Please like Fanpage

(<http://www.facebook.com/programmerthai>)

6 thoughts on “วิธีเขียนโค้ดดึงข้อมูลหุ้นไทย ด้วยภาษา Python (แจกโค้ดฟรี)”

1.  kit พุดว่า:
- 21 มิถุนายน 2017 ที่ 07:59 (<http://www.patanasongsivilai.com/blog/stock-thai-python/#comment-97>)

ตรง getFileName, getStockData นี่ใช้ function อะไรครับ

ตอบกลับ (<http://www.patanasongsivilai.com/blog/stock-thai-python/?replytocom=97#respond>)



แอดมินโฮ โอน้อยออก พูดว่า:

21 มิถุนายน 2017 ที่ 11:42 (<http://www.patanasongsivilai.com/blog/stock-thai-python/#comment-98>)

ขอภัยที่ไม่แสดงโค้ดให้ดูอย่างละเอียด

ลองเข้าไปดูโค้ดเต็มๆ ได้ครับ

https://github.com/adminho/trading-stock-thailand/blob/master/dataset_siamchart.py (https://github.com/adminho/trading-stock-thailand/blob/master/dataset_siamchart.py)

ตอบกลับ (<http://www.patanasongsivilai.com/blog/stock-thai-python/?replytocom=98#respond>)

1.  Yord พูดว่า:

16 กันยายน 2017 ที่ 01:54 (<http://www.patanasongsivilai.com/blog/stock-thai-python/#comment-157>)

เข้าไปที่ https://github.com/adminho/trading-stock-thailand/blob/master/dataset_siamchart.py (https://github.com/adminho/trading-stock-thailand/blob/master/dataset_siamchart.py) แล้วไม่มีไฟล์ รบกวนด้วยครับ

ตอบกลับ (<http://www.patanasongsivilai.com/blog/stock-thai-python/?replytocom=157#respond>)



แอดมินโฮ โอน้อยออก พูดว่า:

21 กันยายน 2017 ที่ 12:22
(<http://www.patanasongsivilai.com/blog/stock-thai-python/#comment-161>)

<https://github.com/adminho/trading-stock-thailand/blob/master/datasets/siamchart.py>
(<https://github.com/adminho/trading-stock-thailand/blob/master/datasets/siamchart.py>)

โทษทีครับ

ตอบกลับ (<http://www.patanasongsivilai.com/blog/stock-thai-python/?replytocom=161#respond>)

2.  **Rawin Ngamloet** (http://twitter.com/rawin_n) พูดว่า:
29 มิถุนายน 2017 ที่ 10:13 (<http://www.patanasongsivilai.com/blog/stock-thai-python/#comment-100>)

บทความดีมากครับ

ปล. ต้องใช้คำว่า “ใส่แห้ง” นะครับ

ตอบกลับ (<http://www.patanasongsivilai.com/blog/stock-thai-python/?replytocom=100#respond>)



แอดมินโฮ โอน้อยออก พูดว่า:

29 มิถุนายน 2017 ที่ 11:26 (<http://www.patanasongsivilai.com/blog/stock-thai-python/#comment-101>)

ขอบคุณครับ

ตอบกลับ (<http://www.patanasongsivilai.com/blog/stock-thai-python/?replytocom=101#respond>)

ใส่ความเห็น

อีเมลของคุณจะไม่แสดงให้คนอื่นเห็น ช่องที่ต้องการถูกทำเครื่องหมาย *

Leave your comment here..*

Your Name*

Email*

Your URL

หมวดหมู่ (Categories)

แสดงความเห็น

Javascript (<http://www.patanasongsivilai.com/blog/category/javascript/>) (1)

AI/Machine Learning (<http://www.patanasongsivilai.com/blog/category/aimachine-learning/>) (8)

ไอที (<http://www.patanasongsivilai.com/blog/category/%e0%b9%84%e0%b8%ad>)

%e0%b8%97%e0%b8%b5/) (37)

เว็บไซต์ (<http://www.patanasongsivilai.com/blog/category/%e0%b9%80%e0%b8%a7%e0%b9%87%e0%b8%9a%e0%b9%84%e0%b8%8b%e0%b8%95%e0%b9%8c/>) (5)

ขายของออนไลน์ (<http://www.patanasongsivilai.com/blog/category/%e0%b8%82%e0%b8%b2%e0%b8%a2%e0%b8%82%e0%b8%ad%e0%b8%87%e0%b8%ad%e0%b8%ad%e0%b8%99%e0%b9%84%e0%b8%a5%e0%b8%99%e0%b9%8c/>) (23)

อื่น ๆ (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/>) (63)

Thailand (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/thailand/>) (1)

ไทย (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/%e0%b9%84%e0%b8%97%e0%b8%a2/>) (4)

ขายของ (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/%e0%b8%82%e0%b8%b2%e0%b8%a2%e0%b8%82%e0%b8%ad%e0%b8%87/>) (2)

พุทธปรัชญา (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/%e0%b8%9e%e0%b8%b8%e0%b8%97%e0%b8%98%e0%b8%9b%e0%b8%a3%e0%b8%b1%e0%b8%8a%e0%b8%8d%e0%b8%b2/>) (12)

สุขภาพ (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/%e0%b8%aa%e0%b8%b8%e0%b8%82%e0%b8%a0%e0%b8%b2%e0%b8%9e/>) (22)

ชายหญิง (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/%e0%b8%8a%e0%b8%b2%e0%b8%a2%e0%b8%ab%e0%b8%8d%e0%b8%b4%e0%b8%87/>) (4)

เพื่อเจ้า (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/%e0%b9%80%e0%b8%9e%e0%b9%89%e0%b8%ad%e0%b9%80%e0%b8%88%e0%b9%89%e0%b8%ad/>) (7)

เงินและการลงทุน (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/%e0%b9%80%e0%b8%87%e0%b8%b4%e0%b8%99%e0%b9%81%e0%b8%a5%e0%b8%b0%e0%b8%81%e0%b8%b2%e0%b8%a3%e0%b8%a5%e0%b8%87%e0%b8%97%e0%b8%b8%e0%b8%99/>) (10)

โหราศาสตร์ (<http://www.patanasongsivilai.com/blog/category/%e0%b8%ad%e0%b8%b7%e0%b9%88%e0%b8%99-%e0%b9%86/%e0%b9%82%e0%b8%ab%e0%b8%a3%e0%b8%b2%e0%b8%a8%e0%b8%b2%e0%b8%aa%e0%b8%95%e0%b8%a3%e0%b9%8c/>) (1)

Uncategorized (<http://www.patanasongsivilai.com/blog/category/uncategorized/>) (2)

Copyright 2016. Your Theme All right Reserved.