U.S. Stock Market V: Shares Outstanding

In [2]:
```
import pandas as pd
import numpy as np
import requests, zipfile, io
import os
from pathlib import Path

from tiingo import TiingoClient
tiingo = TiingoClient({'api_key':'XXXX'})

import matplotlib.pyplot as plt                    # Basic plot library.
plt.style.use('ggplot')                            # Make plots look nice.
```

Get shares:

In [189…
```
# Example:
directory = 'data/sec/merged/'
filename  = '2020q1.csv'
data = pd.read_csv(directory+filename, parse_dates=['filed','ddate'])

tag  = 'EntityCommonStockSharesOutstanding'
item = data[data.tag==tag]

sort = item.sort_values(['cik','filed','ddate'], ascending=[True,True,True])  [:
sort
```

Out[189…

	cik	sic	countryinc	tag	filed	ddate	qtrs	
1670285	1750	3720.0	US	EntityCommonStockSharesOutstanding	2020-03-25	2020-02-29	0	3.51
861534	1800	2834.0	US	EntityCommonStockSharesOutstanding	2020-02-21	2020-01-31	0	1.76
1243206	1961	7372.0	US	EntityCommonStockSharesOutstanding	2020-03-30	2020-03-31	0	5.68

For each firm and each filing date and each ddate, sum all shares and get value column (possibly multiplt share classes):

In [192…
```
sort.groupby(['cik','filed','ddate']).sum()[['value']]
```

Out[192…

			value
cik	**filed**	**ddate**	
1750	**2020-03-25**	**2020-02-29**	3.510070e+07
1800	**2020-02-21**	**2020-01-31**	1.763433e+09
1961	**2020-03-30**	**2020-03-31**	5.681483e+07

Put this into a function:

In [152…

```
def get_shares_from_SEC_files(tags, filename=None):                # Function inp

    directory = 'data/sec/merged/'                                 # Read data fr
    filenames = [filename] if filename else os.listdir(directory) # Supplied fil
    filenames = [f for f in filenames if not f.startswith(".")]   # Exclude hidd

    results   = {t:pd.DataFrame() for t in tags}                   # Dictionary o

    for filename in filenames:                                     # Loop over al
        print(filename)
        data = pd.read_csv(directory+filename, parse_dates=['filed','ddate'])  #

        for t in tags:                                             # Loop over al
            item   = data[data.tag==t]                              # Select all
            sort   = item.sort_values(['cik','filed','ddate'], ascending=[True,T
            total  = sort.groupby(['cik','filed','ddate']).sum()
            latest = total.groupby(['cik','filed']).last()
            results[t] = results[t].append( latest )

    for t in tags:                                                 # Now sort all
        if not results[t].empty: results[t] = results[t].sort_index(level='filed

    return results
```

In [194…

```
tags = ['EntityCommonStockSharesOutstanding','CommonStockSharesOutstanding']

items = get_shares_from_SEC_files(tags)
```

```
2018q4.csv
2018q3.csv
2018q2.csv
2021_01.csv
2018q1.csv
2020q2.csv
2020q3.csv
2020q1.csv
2019q4.csv
2019q1.csv
2019q3.csv
2019q2.csv
2013q4.csv
2015q2.csv
2015q3.csv
2017q1.csv
2020_12.csv
2020_10.csv
2017q3.csv
2015q1.csv
2017q2.csv
2011q4.csv
2020_11.csv
2013q2.csv
2015q4.csv
2011q1.csv
2013q3.csv
2013q1.csv
2011q3.csv
2017q4.csv
2011q2.csv
2009q4.csv
2014q1.csv
```

```
2016q3.csv
2010q4.csv
2016q2.csv
2014q2.csv
2012q4.csv
2016q1.csv
2014q3.csv
2009q2.csv
2010q3.csv
2012q1.csv
2010q2.csv
2016q4.csv
2009q3.csv
2014q4.csv
2012q2.csv
2012q3.csv
2010q1.csv
```

EntityCommonStockSharesOutstanding:

In [195…
```
items['EntityCommonStockSharesOutstanding']   [-4:]
```

Out[195…

		value
cik	**filed**	
1551887	**2021-01-29**	43892801.0
1580149	**2021-01-29**	13916164.0
1593812	**2021-01-29**	0.0
1807707	**2021-01-29**	12650000.0

Set zero shares to missing:

In [201…
```
items['EntityCommonStockSharesOutstanding'] = items['EntityCommonStockSharesOuts
items['CommonStockSharesOutstanding']       = items['CommonStockSharesOutstandin
```

Combine these two tags (replace missing EntityCommonStockSharesOutstanding with CommonStockSharesOutstanding):

In [202…
```
def combine_items(tags, items):
    result = items[tags[0]]
    for tag in tags[1:]:  result = result.combine_first( items[tag] )
    return result


shares = combine_items(tags, items)
shares
```

Out[202…

		value
cik	**filed**	
1750	**2010-09-23**	39662816.0
	2010-12-21	39677152.0
	2011-03-22	39739031.0

		value
cik	**filed**	
	2011-07-13	39682142.0
	2011-09-23	40460631.0
...	...	...
1822835	**2020-12-22**	5750000.0
1822929	**2020-12-07**	13459124.0
1823365	**2020-12-23**	2463507.0
1823383	**2021-01-22**	15050833.0
1823882	**2020-12-14**	14920000.0

234542 rows × 1 columns

In [203…
```
# Save this table:
shares.to_csv('data/sec/items/SharesOutstanding.csv')
```

In [204…
```
# Read the table we just saved:
shares = pd.read_csv('data/sec/items/SharesOutstanding.csv', parse_dates=['filed
shares
```

Out[204…

		value
cik	**filed**	
1750	**2010-09-23**	39662816.0
	2010-12-21	39677152.0
	2011-03-22	39739031.0
	2011-07-13	39682142.0
	2011-09-23	40460631.0
...	...	...
1822835	**2020-12-22**	5750000.0
1822929	**2020-12-07**	13459124.0
1823365	**2020-12-23**	2463507.0
1823383	**2021-01-22**	15050833.0
1823882	**2020-12-14**	14920000.0

234542 rows × 1 columns

Forward-fill shares to all trading days:

In [205…
```
def ffill_values(item, dates):
    data = item.unstack('cik')
    data = data.reindex(dates.union(data.index)).sort_index()            # Add sp
```

```
    filing_dates = pd.read_csv('data/sec/dates/filing_dates.csv', index_col='cik
    last_filing_date_all_firms = filing_dates.max()                        # Most r

    for cik in data.columns:                                               # Loop o
        last_filing_date      = pd.Series(filing_dates[cik]).iloc[-1]   # Last d
        days_since_last_filed = (last_filing_date_all_firms - last_filing_date).
        last_date_this_firm   = dates[-1] if days_since_last_filed < 120 else la
        data.loc[:last_date_this_firm, cik].ffill(inplace=True)            # Forwar

    return data.loc[dates]                                                 # Return



trading_days = pd.to_datetime( tiingo.get_dataframe('SPY','2009-04-15').index ).

shares = ffill_values(shares.value, trading_days)
shares
```

Out[205…

cik	1750	1800	1961	2034	2098	2178	2186	2
date								
2009-04-15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2009-04-16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2009-04-17	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2009-04-20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2009-04-21	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	
2021-03-04	35292336.0	1.772362e+09	56814833.0	NaN	3338913.0	4242284.0	12511966.0	1.211280
2021-03-05	35292336.0	1.772362e+09	56814833.0	NaN	3338913.0	4242284.0	12511966.0	1.211280
2021-03-08	35292336.0	1.772362e+09	56814833.0	NaN	3338913.0	4242284.0	12511966.0	1.211280
2021-03-09	35292336.0	1.772362e+09	56814833.0	NaN	3338913.0	4242284.0	12511966.0	1.211280
2021-03-10	35292336.0	1.772362e+09	56814833.0	NaN	3338913.0	4242284.0	12511966.0	1.211280

2997 rows × 11590 columns

Shares outstanding for specific firm:

In [206…

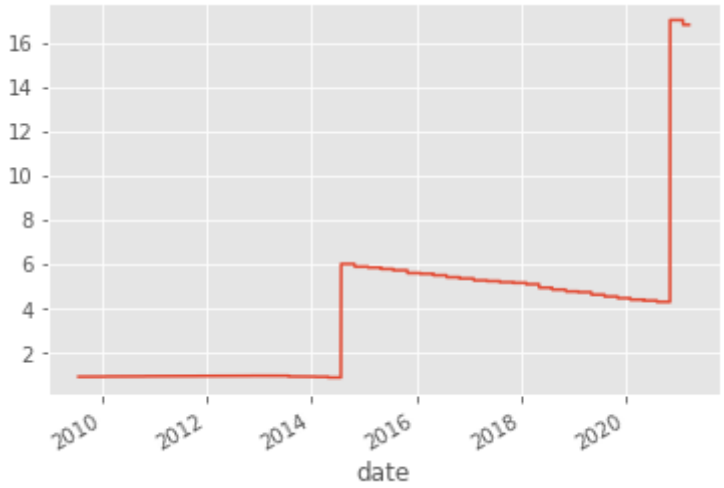
```
symbols = pd.read_csv('data/ticker_symbols/symbols.csv',index_col=0)
```

In [207…

```
cik = symbols[symbols.ticker=='AAPL'].index[0]
```

```
shares[cik].div(10**9).plot()
```

Out[207… <AxesSubplot:xlabel='date'>



Get some prices:

In [252…

```
close = tiingo.get_dataframe(['AAPL','MSFT','AMZN','TSLA'],'2009-01-01', metric_
close.index = pd.to_datetime(close.index).tz_convert(None)
close
```

Out[252…

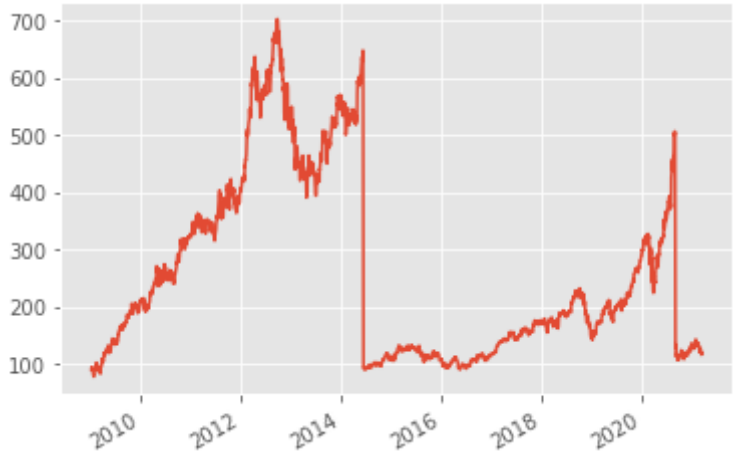
	AAPL	MSFT	AMZN	TSLA
2009-01-02	90.750	20.33	54.36	NaN
2009-01-05	94.580	20.52	54.06	NaN
2009-01-06	93.020	20.76	57.36	NaN
2009-01-07	91.010	19.51	56.20	NaN
2009-01-08	92.700	20.12	57.16	NaN
…	…	…	…	…
2021-03-04	120.130	226.73	2977.57	621.44
2021-03-05	121.420	231.60	3000.46	597.95
2021-03-08	116.360	227.39	2951.95	563.00
2021-03-09	121.085	233.78	3062.85	673.58
2021-03-10	119.980	232.42	3057.64	668.06

3067 rows × 4 columns

In [210…

```
close['AAPL'].plot()
```

Out[210… <AxesSubplot:>



Calculate market cap for single firm:

In [257…
```
shares = shares.rename( columns=symbols.ticker )
shares[-3:]
```

Out[257…

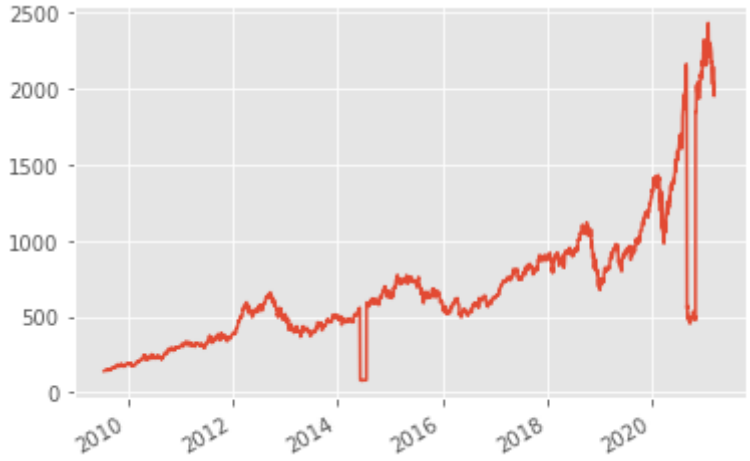
cik	AIR	ABT	WDDD	2034	ACU	AE	BKTI	
date								
2021-03-08	35292336.0	1.772362e+09	56814833.0	NaN	3338913.0	4242284.0	12511966.0	1.211280e
2021-03-09	35292336.0	1.772362e+09	56814833.0	NaN	3338913.0	4242284.0	12511966.0	1.211280e
2021-03-10	35292336.0	1.772362e+09	56814833.0	NaN	3338913.0	4242284.0	12511966.0	1.211280e

3 rows × 11590 columns

In [258…
```
ticker = 'AAPL'

(shares[ticker] * close[ticker]).div(10**9).plot()
```

Out[258… <AxesSubplot:>

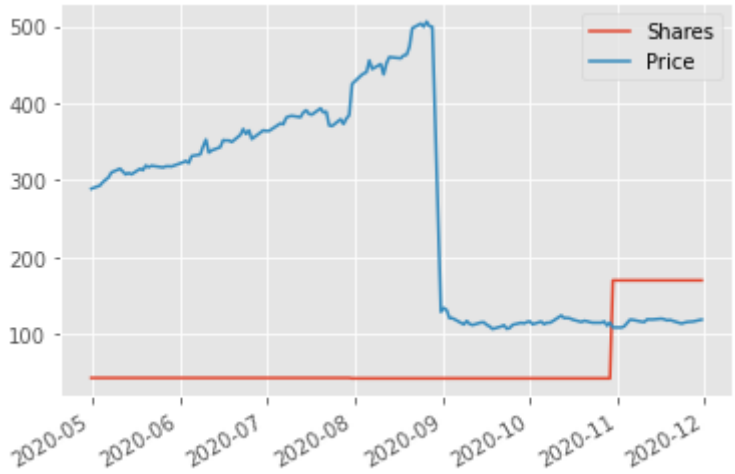


What is going on here?

Compare shares and price of Apple from 2020-5 to 2020-11:

In [259…
```
pd.DataFrame({'Shares':shares[ticker]/10**8, 'Price':close[ticker]}).loc['2020-5
```

Out[259…
```
<AxesSubplot:>
```



Get stock splits for these firms:

In [181…
```
splitFactor = tiingo.get_dataframe(['AAPL','MSFT','AMZN','TSLA'], '2009-01-01',
splitFactor.index = pd.to_datetime(splitFactor.index).tz_convert(None)
splitFactor
```

Out[181…

	AAPL	MSFT	AMZN	TSLA
2009-01-02	1.0	1.0	1.0	NaN
2009-01-05	1.0	1.0	1.0	NaN
2009-01-06	1.0	1.0	1.0	NaN
2009-01-07	1.0	1.0	1.0	NaN
2009-01-08	1.0	1.0	1.0	NaN
...	...	...	...	...
2021-03-04	1.0	1.0	1.0	1.0

	AAPL	MSFT	AMZN	TSLA
2021-03-05	1.0	1.0	1.0	1.0
2021-03-08	1.0	1.0	1.0	1.0
2021-03-09	1.0	1.0	1.0	1.0
2021-03-10	1.0	1.0	1.0	1.0

3067 rows × 4 columns

Generate table with split adjustments:

In [260…

```
filing_dates = pd.read_csv('data/sec/dates/filing_dates.csv', index_col='cik', p

ones = pd.DataFrame({'date':filing_dates, 'one':1})
ones = ones.set_index([ones.index,ones.date]).one.unstack('cik')
ones = ones.rename( columns=symbols.ticker )
ones = ones.reindex( trading_days.union(ones.index) )

split_adjustments = ones.fillna( splitFactor[splitFactor!=1] )
split_adjustments = split_adjustments.ffill()

split_adjustments
```

Out[260…

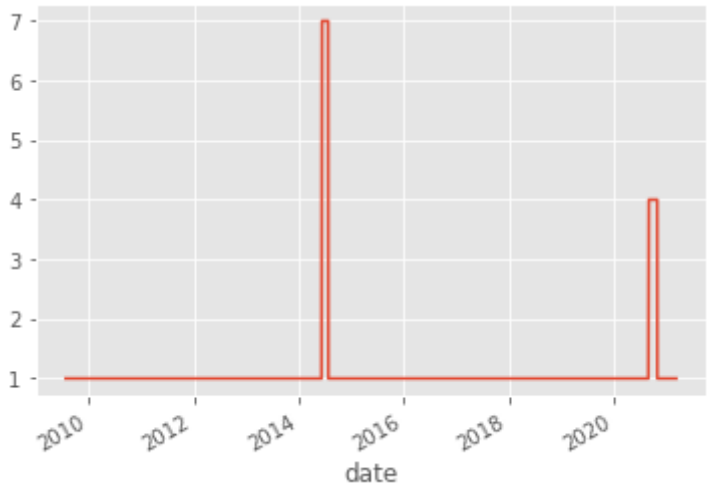
cik date	AIR	ABT	WDDD	2034	ACU	AE	BKTI	AMD	2491	APD	…	HAACU	CND	ACIC	
2009-04-15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	…	NaN	NaN	NaN	
2009-04-16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	…	NaN	NaN	NaN	
2009-04-17	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	…	NaN	NaN	NaN	
2009-04-20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	…	NaN	NaN	NaN	
2009-04-21	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	…	NaN	NaN	NaN	
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	
2021-03-04	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	…	1.0	1.0	1.0	
2021-03-05	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	…	1.0	1.0	1.0	
2021-03-08	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	…	1.0	1.0	1.0	
2021-03-09	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	…	1.0	1.0	1.0	
2021-03-10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	…	1.0	1.0	1.0	

3011 rows × 12126 columns

Plot these values for specific firm:

In [261…
```
split_adjustments['AAPL'].plot()
```

Out[261…
```
<AxesSubplot:xlabel='date'>
```

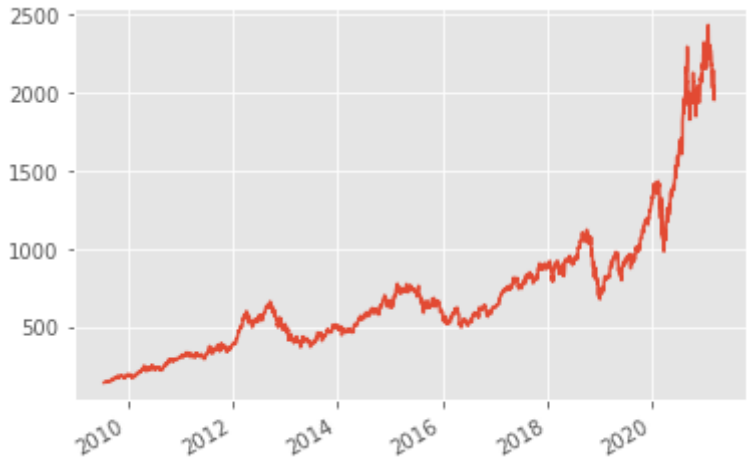


Market cap for this firm:

In [256…
```
ticker = 'AAPL'
cik    = symbols[symbols.ticker==ticker].index[0]

(shares[cik] * split_adjustments[ticker] * close[ticker]).div(10**9).plot()
```

Out[256…
```
<AxesSubplot:>
```



Caculate market cap for all firms:

In [262…
```
mcap = shares * split_adjustments * close
mcap
```

Out[262…

	2034	2491	3116	3673	3952	3982	4187	4515	4611	4828	...	ZTS	ZUMZ	ZUO
2009-01-02	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN

	2034	2491	3116	3673	3952	3982	4187	4515	4611	4828	...	ZTS	ZUMZ	ZUO
2009-01-05	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
2009-01-06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
2009-01-07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
2009-01-08	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2021-03-04	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
2021-03-05	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
2021-03-08	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
2021-03-09	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
2021-03-10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN

3081 rows × 12126 columns

Check market cap of specific firm:

In [263…
```
mcap['AAPL'].plot()
```

Out[263… <AxesSubplot:>

Get sales and earnings:

In [62]:
```
trading_days = tiingo.get_dataframe('SPY','2009-04-15').index.tz_convert(None)

sales    = pd.read_csv('data/sec/items/Sales.csv',    parse_dates=['filed'], ind
earnings = pd.read_csv('data/sec/items/Earnings.csv', parse_dates=['filed'], ind
```

```
salesQ = ffill_values(sales.valueQ, trading_days)
salesA = ffill_values(sales.valueA, trading_days)

earningsQ = ffill_values( earnings.valueQ, trading_days )
earningsA = ffill_values( earnings.valueA, trading_days )
```

Price-earnings ratio

In [237…

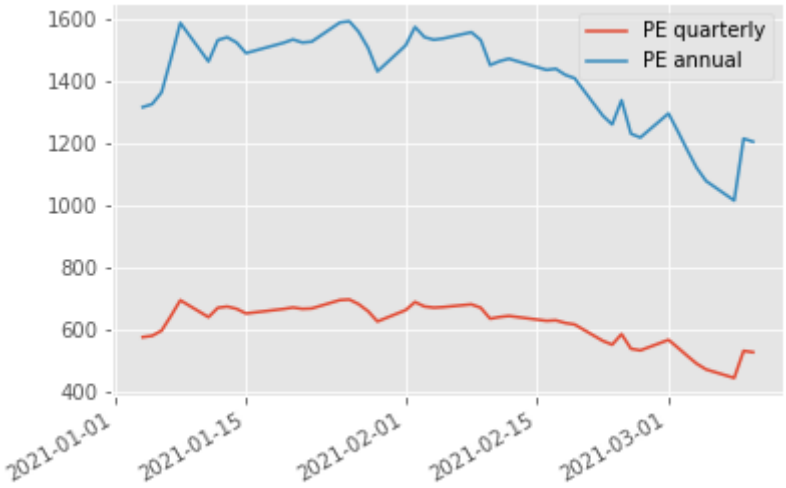
```
ticker = 'TSLA'

cik = symbols[symbols.ticker==ticker].index[0]

t = pd.DataFrame()
t['PE quarterly'] = mcap[ticker] / (earningsQ[cik] * 4)
t['PE annual']    = mcap[ticker] /  earningsA[cik]

t['2021':].plot()
```

Out[237… <AxesSubplot:>



Price to sales:

In [238…

```
ticker = 'TSLA'

cik = symbols[symbols.ticker==ticker].index[0]

t = pd.DataFrame()
t['PS quarterly'] = mcap[ticker] / (salesQ[cik] * 4)
t['PS annual']    = mcap[ticker] /  salesA[cik]

t['2015':].plot()
```

Out[238… <AxesSubplot:>


PS quarterly
PS annual
30
25
20
15
10
5
0
2015
2016
2017
2018
2019
2020
2021

In [ ]: