

# Stock Prices I

```
In [1]: import pandas as pd

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

Get data for specific ticker symbol:

```
In [2]: tiingo.get_dataframe('MSFT', '2010-1-1')
```

```
Out[2]:
```

	close	high	low	open	volume	adjClose	adjHigh	a
date								
2010-01-04 00:00:00+00:00	30.950	31.1000	30.59	30.620	38409100	24.051848	24.168416	23.7
2010-01-05 00:00:00+00:00	30.960	31.1000	30.64	30.850	49749600	24.059620	24.168416	23.8
2010-01-06 00:00:00+00:00	30.770	31.0800	30.52	30.880	58182400	23.911967	24.152874	23.7
2010-01-07 00:00:00+00:00	30.452	30.7000	30.19	30.630	50559700	23.664843	23.857569	23.4
2010-01-08 00:00:00+00:00	30.660	30.8800	30.24	30.280	51197400	23.826484	23.997450	23.5
...	...	...	...	...	...	...	...	...
2021-02-24 00:00:00+00:00	234.550	235.2000	229.00	230.010	26339746	234.550000	235.200000	229.0
2021-02-25 00:00:00+00:00	228.990	234.5852	227.88	232.080	39118089	228.990000	234.585200	227.8
2021-02-26 00:00:00+00:00	232.380	235.3700	229.54	231.525	37545055	232.380000	235.370000	229.5
2021-03-01 00:00:00+00:00	236.940	237.4700	233.15	235.900	25332837	236.940000	237.470000	233.1
2021-03-02 00:00:00+00:00	233.870	237.3000	233.45	237.010	22209828	233.870000	237.300000	233.4

2809 rows × 12 columns

Get specific metric for multiple ticker symbols:

```
In [3]: tiingo.get_dataframe(['AAPL', 'MSFT'], '2010-1-1', metric_name='close')
```

```
Out[3]:
```

	AAPL	MSFT
2010-01-04 00:00:00+00:00	214.01	30.950
2010-01-05 00:00:00+00:00	214.38	30.960

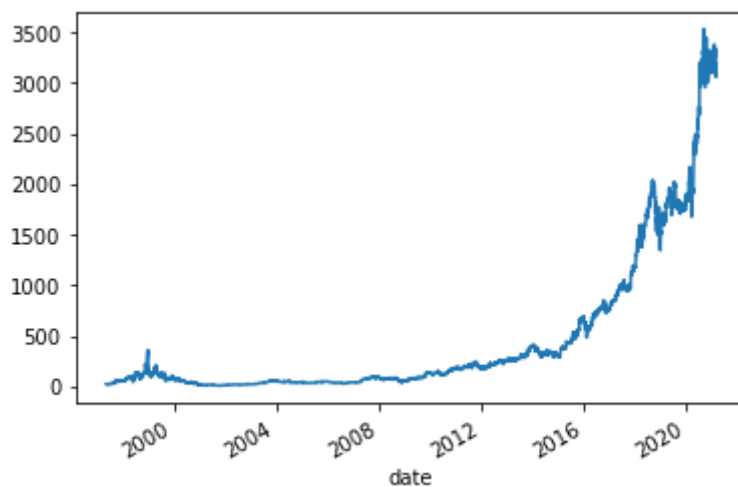
	AAPL	MSFT
2010-01-06 00:00:00+00:00	210.97	30.770
2010-01-07 00:00:00+00:00	210.58	30.452
2010-01-08 00:00:00+00:00	211.98	30.660
...	...	...
2021-02-24 00:00:00+00:00	125.35	234.550
2021-02-25 00:00:00+00:00	120.99	228.990
2021-02-26 00:00:00+00:00	121.26	232.380
2021-03-01 00:00:00+00:00	127.79	236.940
2021-03-02 00:00:00+00:00	125.12	233.870

2809 rows × 2 columns

Plot Amazon close:

```
In [4]: amzn = tiingo.get_dataframe('AMZN','1900-1-1')
        amzn.close.plot()
```

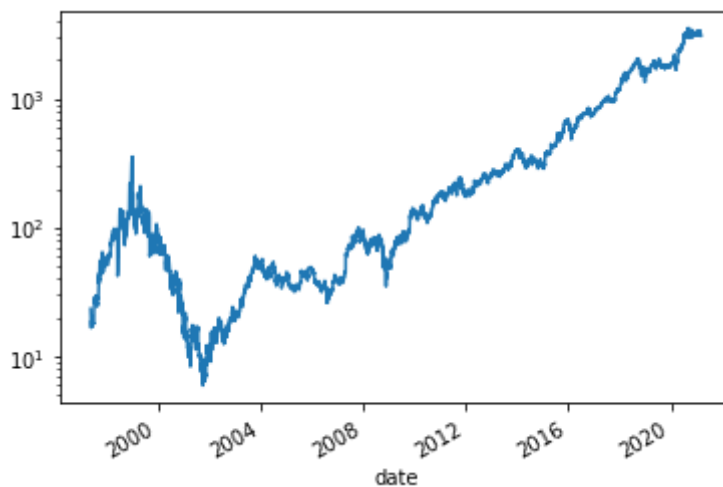
```
Out[4]: <AxesSubplot:xlabel='date'>
```



Plot on log scale to see the relative prices changes better:

```
In [5]: amzn.close.plot(logy=True)
```

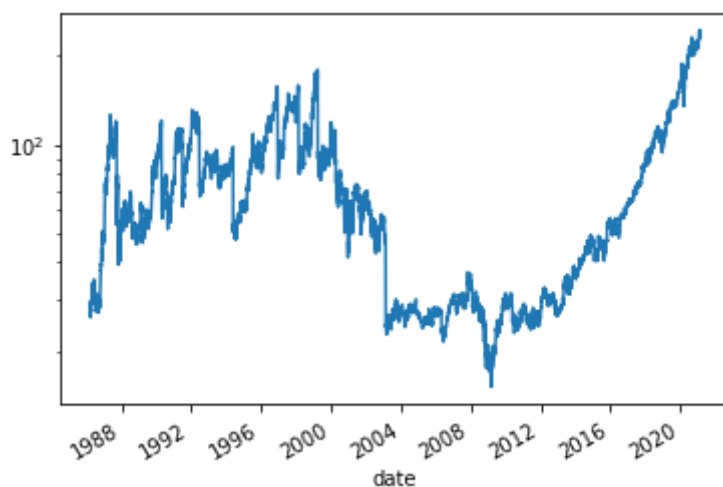
```
Out[5]: <AxesSubplot:xlabel='date'>
```



Plot Microsoft close:

```
In [6]: data = tiingo.get_dataframe('MSFT', '1980-1-1')
data.close.plot(logy=True)
```

Out[6]: <AxesSubplot:xlabel='date'>



Get dates where Microsoft had a stock split:

```
In [7]: data[data.splitFactor>1]
```

Out[7]:

	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjC
1987-09-21 00:00:00+00:00	53.50	57.25	53.00	53.50	297044	0.236491	0.253067	0.234281	0.236
1990-04-16 00:00:00+00:00	60.75	61.75	60.13	61.00	274089	0.537078	0.545919	0.531596	0.539
1991-06-27 00:00:00+00:00	68.00	68.25	66.75	67.75	784844	0.901760	0.905075	0.885184	0.898
1992-06-15 00:00:00+00:00	75.75	78.00	75.00	75.00	1144600	1.506801	1.551558	1.491883	1.491

	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjC
date									
1994-05-23 00:00:00+00:00	50.56	51.00	48.50	49.00	2342100	2.011456	2.028960	1.929501	1.949
1996-12-09 00:00:00+00:00	81.75	81.87	78.00	78.37	5919900	6.504608	6.514156	6.206231	6.23!
1998-02-23 00:00:00+00:00	81.62	81.69	79.37	80.94	15100450	12.988528	12.999668	12.630477	12.88!
1999-03-29 00:00:00+00:00	92.37	92.62	87.87	90.12	19944250	29.398441	29.478008	27.966234	28.68!
2003-02-18 00:00:00+00:00	24.96	24.99	24.40	24.62	28707750	15.887952	15.907048	15.531492	15.67!

If you bought 1 share at Microsoft's IPO, how many shares do you have now?

```
In [8]: data.splitFactor[data.splitFactor>1].prod()
```

```
Out[8]: 288.0
```

```
In [9]: data.splitFactor[data.splitFactor>1].cumprod() # Note the difference between p
```

```
Out[9]: date
1987-09-21 00:00:00+00:00      2.0
1990-04-16 00:00:00+00:00      4.0
1991-06-27 00:00:00+00:00      6.0
1992-06-15 00:00:00+00:00      9.0
1994-05-23 00:00:00+00:00     18.0
1996-12-09 00:00:00+00:00     36.0
1998-02-23 00:00:00+00:00     72.0
1999-03-29 00:00:00+00:00    144.0
2003-02-18 00:00:00+00:00    288.0
Name: splitFactor, dtype: float64
```

Get all dates where Microsoft paid a dividend:

```
In [10]: dividends = data.loc[data.divCash>0,['close', 'divCash']]
dividends
```

```
Out[10]:
```

	close	divCash
date		
2003-02-19 00:00:00+00:00	24.53	0.08
2003-10-15 00:00:00+00:00	29.07	0.16
2004-08-23 00:00:00+00:00	27.24	0.08
2004-11-15 00:00:00+00:00	27.39	3.08
2005-02-15 00:00:00+00:00	25.93	0.08
...	...	...
2020-02-19 00:00:00+00:00	187.28	0.51

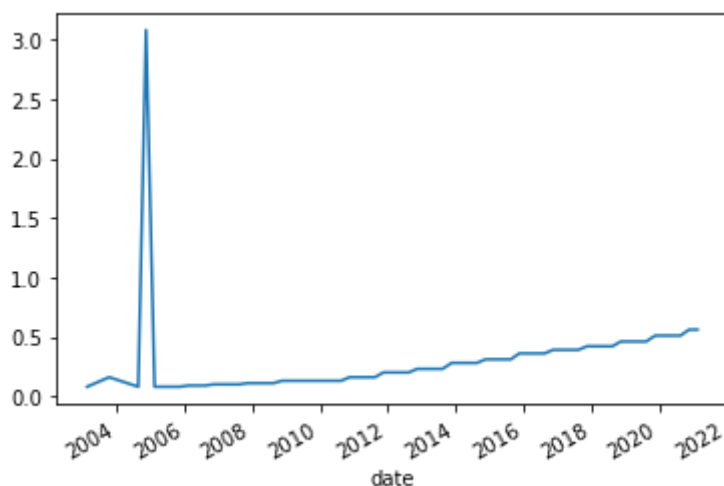
	close	divCash
date		
2020-05-20 00:00:00+00:00	185.66	0.51
2020-08-19 00:00:00+00:00	209.70	0.51
2020-11-18 00:00:00+00:00	211.08	0.56
2021-02-17 00:00:00+00:00	244.20	0.56

69 rows × 2 columns

Plot the dividends:

```
In [11]: dividends.divCash.plot()
```

```
Out[11]: <AxesSubplot:xlabel='date'>
```



Calculate dividend yield:

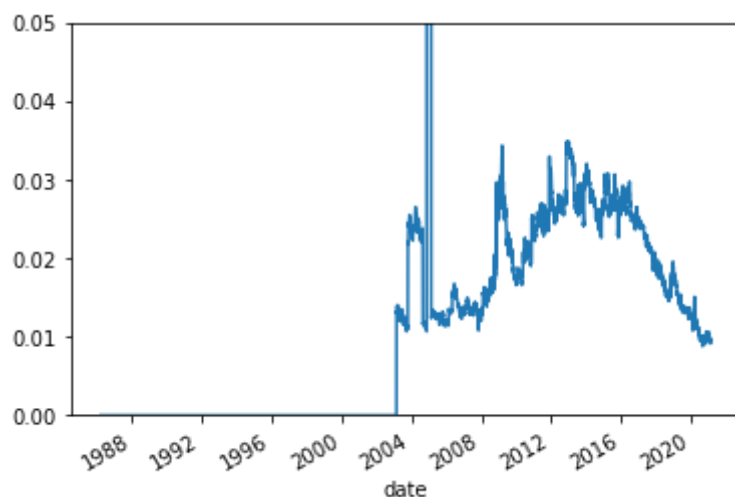
```
In [12]: # Fill days with 0 dividend with previously paid dividend -> multiply by 4 to an
divided_yield = data.divCash.replace(0, method='ffill').multiply(4).div(data.close)
divided_yield
```

```
Out[12]: date
1986-03-13 00:00:00+00:00    0.000000
1986-03-14 00:00:00+00:00    0.000000
1986-03-17 00:00:00+00:00    0.000000
1986-03-18 00:00:00+00:00    0.000000
1986-03-19 00:00:00+00:00    0.000000
...
2021-02-24 00:00:00+00:00    0.009550
2021-02-25 00:00:00+00:00    0.009782
2021-02-26 00:00:00+00:00    0.009639
2021-03-01 00:00:00+00:00    0.009454
2021-03-02 00:00:00+00:00    0.009578
Length: 8814, dtype: float64
```

Plot this:

```
In [13]: divided_yield.plot(ylim=(0,0.05))
```

Out[13]: <AxesSubplot:xlabel='date'>



Microsoft in November 2004:

In [14]: `data.loc['2004-11']` # Note the price drop on 2004-11-15 due to special dividend

Out[14]:

	close	high	low	open	volume	adjClose	adjHigh	adjLow	a
date									
2004-11-01 00:00:00+00:00	28.08	28.28	27.960	28.150	72930900	18.083892	18.212694	18.006610	18.
2004-11-02 00:00:00+00:00	28.24	28.47	28.029	28.240	89417100	18.186934	18.335057	18.051047	18.
2004-11-03 00:00:00+00:00	28.47	28.65	28.310	28.650	79666700	18.335057	18.450979	18.232015	18.
2004-11-04 00:00:00+00:00	29.00	29.00	28.380	28.390	87867700	18.676384	18.676384	18.277095	18.
2004-11-05 00:00:00+00:00	29.31	29.36	29.030	29.160	95337700	18.876028	18.908228	18.695704	18.
2004-11-08 00:00:00+00:00	29.28	29.48	29.130	29.190	112802100	18.856707	18.985510	18.760105	18.
2004-11-09 00:00:00+00:00	29.77	29.89	29.350	29.430	100401000	19.172274	19.249555	18.901788	18.
2004-11-10 00:00:00+00:00	29.73	30.00	29.690	29.910	84097700	19.146513	19.320397	19.120753	19.
2004-11-11 00:00:00+00:00	29.98	30.08	29.820	29.885	87358900	19.307517	19.371918	19.204475	19.
2004-11-12 00:00:00+00:00	29.97	30.20	29.800	30.160	162269000	19.301077	19.449200	19.191594	19.
2004-11-15 00:00:00+00:00	27.39	27.50	27.200	27.350	104468000	19.623083	19.701891	19.486961	19.
2004-11-16 00:00:00+00:00	27.12	27.34	27.050	27.320	64522600	19.429646	19.587262	19.379496	19.

	close	high	low	open	volume	adjClose	adjHigh	adjLow	a
date									
2004-11-17 00:00:00+00:00	27.17	27.36	27.060	27.250	58830700	19.465468	19.601590	19.386660	19.
2004-11-18 00:00:00+00:00	27.07	27.17	27.000	27.130	63249900	19.393825	19.465468	19.343675	19
2004-11-19 00:00:00+00:00	26.86	27.07	26.840	27.040	85808600	19.243374	19.393825	19.229045	19.
2004-11-22 00:00:00+00:00	26.65	26.82	24.860	26.760	92410800	19.092923	19.214717	17.810509	19
2004-11-23 00:00:00+00:00	26.53	26.70	26.400	26.520	70459700	19.006951	19.128745	18.913815	18.
2004-11-24 00:00:00+00:00	26.64	26.73	26.400	26.620	60069200	19.085759	19.150238	18.913815	19
2004-11-26 00:00:00+00:00	26.60	26.82	26.550	26.570	24398700	19.057102	19.214717	19.021280	19.
2004-11-29 00:00:00+00:00	26.77	26.95	26.610	26.635	67079900	19.178895	19.307853	19.064266	19
2004-11-30 00:00:00+00:00	26.81	27.01	26.700	26.760	75960400	19.207552	19.350839	19.128745	19

Microsoft in February 2003:

In [15]:

```
data.loc['2003-2'] # Note the price drop on 2003-02-18 due to stock split
```

Out[15]:

	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjC
date									
2003-02-03 00:00:00+00:00	48.56	49.10	47.46	47.93	40064900	15.455108	15.626972	15.105012	15.254
2003-02-04 00:00:00+00:00	47.32	47.94	46.88	47.80	40927400	15.060455	15.257781	14.920417	15.213
2003-02-05 00:00:00+00:00	46.96	48.53	46.73	47.83	50536000	14.945878	15.445559	14.872677	15.223
2003-02-06 00:00:00+00:00	47.42	47.62	46.56	46.86	41656000	15.092282	15.155935	14.818571	14.914
2003-02-07 00:00:00+00:00	46.58	47.95	46.41	47.88	37604000	14.824936	15.260964	14.770831	15.238
2003-02-10 00:00:00+00:00	47.38	47.52	46.54	46.80	37040700	15.079551	15.124109	14.812206	14.894
2003-02-11 00:00:00+00:00	46.44	47.67	46.00	47.30	42146000	14.780379	15.171849	14.640341	15.054
2003-02-12 00:00:00+00:00	46.44	47.10	46.26	46.55	35657600	14.780379	14.990436	14.723091	14.815
2003-02-13 00:00:00+00:00	46.99	47.12	46.13	46.41	36779100	14.955426	14.996801	14.681716	14.770

	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjC
date									
2003-02-14 00:00:00+00:00	48.30	48.50	46.77	47.25	45223200	15.372358	15.436011	14.885407	15.038
2003-02-18 00:00:00+00:00	24.96	24.99	24.40	24.62	28707750	15.887952	15.907048	15.531492	15.671
2003-02-19 00:00:00+00:00	24.53	24.88	24.17	24.82	46902700	15.665165	15.888679	15.435264	15.850
2003-02-20 00:00:00+00:00	24.14	24.87	24.10	24.77	50897200	15.416106	15.882293	15.390561	15.818
2003-02-21 00:00:00+00:00	24.63	24.80	23.70	24.29	56853200	15.729026	15.837590	15.135116	15.511
2003-02-24 00:00:00+00:00	24.07	24.50	23.84	24.44	62403700	15.371403	15.646006	15.224522	15.607
2003-02-25 00:00:00+00:00	24.19	24.27	23.38	23.54	68113000	15.448036	15.499125	14.930760	15.032
2003-02-26 00:00:00+00:00	23.61	24.47	23.58	24.07	57096000	15.077641	15.626848	15.058483	15.371
2003-02-27 00:00:00+00:00	23.58	24.21	23.30	23.90	75434300	15.058483	15.460809	14.879671	15.262
2003-02-28 00:00:00+00:00	23.70	24.07	23.55	23.74	56585400	15.135116	15.371403	15.039324	15.160

Add column for % change of close price:

In [16]:

```
data['close_pct_change'] = data.close.pct_change()
data['2003-2-10': '2003-2-20']
```

Out[16]:

	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjC
date									
2003-02-10 00:00:00+00:00	47.38	47.52	46.54	46.80	37040700	15.079551	15.124109	14.812206	14.894
2003-02-11 00:00:00+00:00	46.44	47.67	46.00	47.30	42146000	14.780379	15.171849	14.640341	15.054
2003-02-12 00:00:00+00:00	46.44	47.10	46.26	46.55	35657600	14.780379	14.990436	14.723091	14.815
2003-02-13 00:00:00+00:00	46.99	47.12	46.13	46.41	36779100	14.955426	14.996801	14.681716	14.770
2003-02-14 00:00:00+00:00	48.30	48.50	46.77	47.25	45223200	15.372358	15.436011	14.885407	15.038
2003-02-18 00:00:00+00:00	24.96	24.99	24.40	24.62	28707750	15.887952	15.907048	15.531492	15.671
2003-02-19 00:00:00+00:00	24.53	24.88	24.17	24.82	46902700	15.665165	15.888679	15.435264	15.850
2003-02-20 00:00:00+00:00	24.14	24.87	24.10	24.77	50897200	15.416106	15.882293	15.390561	15.818



Add column for % change of adjusted close price:

```
In [17]: data['adjClose_pct_change'] = data.adjClose.pct_change()  
data['2003-2-10':'2003-2-20']
```

```
Out[17]:
```

	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjC
date									
2003-02-10 00:00:00+00:00	47.38	47.52	46.54	46.80	37040700	15.079551	15.124109	14.812206	14.894
2003-02-11 00:00:00+00:00	46.44	47.67	46.00	47.30	42146000	14.780379	15.171849	14.640341	15.054
2003-02-12 00:00:00+00:00	46.44	47.10	46.26	46.55	35657600	14.780379	14.990436	14.723091	14.815
2003-02-13 00:00:00+00:00	46.99	47.12	46.13	46.41	36779100	14.955426	14.996801	14.681716	14.770
2003-02-14 00:00:00+00:00	48.30	48.50	46.77	47.25	45223200	15.372358	15.436011	14.885407	15.038
2003-02-18 00:00:00+00:00	24.96	24.99	24.40	24.62	28707750	15.887952	15.907048	15.531492	15.671
2003-02-19 00:00:00+00:00	24.53	24.88	24.17	24.82	46902700	15.665165	15.888679	15.435264	15.850
2003-02-20 00:00:00+00:00	24.14	24.87	24.10	24.77	50897200	15.416106	15.882293	15.390561	15.818

Return on 2003-02-18:

```
In [19]: 24.96 * 2 / 48.30 - 1.    # * 2: stock split
```

```
Out[19]: 0.03354037267080745
```

Note how this return is equal to % change of adjClose (last column).

Compare close, adjusted close for last 15 days:

```
In [20]: data[['close', 'adjClose', 'divCash', 'splitFactor', 'close_pct_change', 'adjClose_pct_change']]
```

```
Out[20]:
```

	close	adjClose	divCash	splitFactor	close_pct_change	adjClose_pct_change
date						
2021-02-09 00:00:00+00:00	243.77	243.212265	0.00	1.0	0.005361	0.0053
2021-02-10 00:00:00+00:00	242.82	242.264439	0.00	1.0	-0.003897	-0.0038
2021-02-11 00:00:00+00:00	244.49	243.930618	0.00	1.0	0.006878	0.0068
2021-02-12 00:00:00+00:00	244.99	244.429474	0.00	1.0	0.002045	0.0020

	close	adjClose	divCash	splitFactor	close_pct_change	adjClose_pct_chan
date						
2021-02-16 00:00:00+00:00	243.70	243.142425	0.00	1.0	-0.005266	-0.005266
2021-02-17 00:00:00+00:00	244.20	244.200000	0.56	1.0	0.002052	0.004308
2021-02-18 00:00:00+00:00	243.79	243.790000	0.00	1.0	-0.001679	-0.001679
2021-02-19 00:00:00+00:00	240.97	240.970000	0.00	1.0	-0.011567	-0.011567
2021-02-22 00:00:00+00:00	234.51	234.510000	0.00	1.0	-0.026808	-0.026808
2021-02-23 00:00:00+00:00	233.27	233.270000	0.00	1.0	-0.005288	-0.005288
2021-02-24 00:00:00+00:00	234.55	234.550000	0.00	1.0	0.005487	0.005487
2021-02-25 00:00:00+00:00	228.99	228.990000	0.00	1.0	-0.023705	-0.023705
2021-02-26 00:00:00+00:00	232.38	232.380000	0.00	1.0	0.014804	0.014804
2021-03-01 00:00:00+00:00	236.94	236.940000	0.00	1.0	0.019623	0.019623
2021-03-02 00:00:00+00:00	233.87	233.870000	0.00	1.0	-0.012957	-0.012957

Compare returns:

```
In [21]: 244.20 / 243.70 - 1 # Close prices
```

```
Out[21]: 0.002051702913418163
```

```
In [22]: (244.20 + 0.56) / 243.70 - 1 # Close prices + dividend
```

```
Out[22]: 0.004349610176446417
```

```
In [23]: 244.20 / 243.142425 - 1 # Adjusted close prices
```

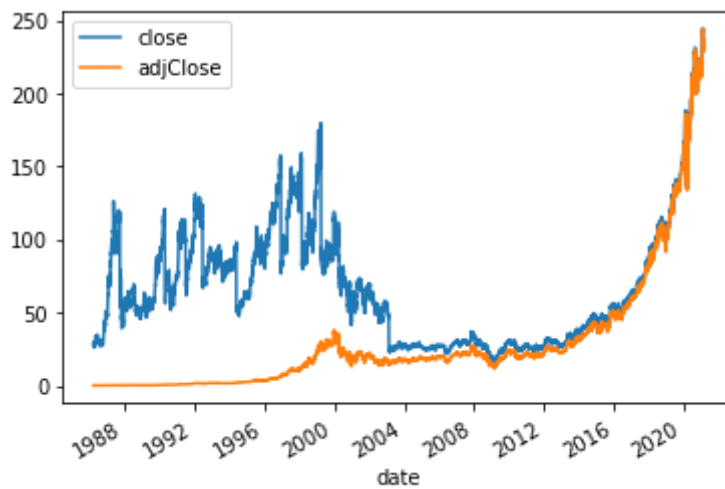
```
Out[23]: 0.004349611138409815
```

Note how the % change of the adjusted close is equal to the return including dividend.

Plot close vs adjClose:

```
In [24]: data[['close', 'adjClose']].plot()
```

```
Out[24]: <AxesSubplot:xlabel='date'>
```



Daily returns:

```
In [25]: r = data.adjClose.pct_change()
         r
```

```
Out[25]: date
1986-03-13 00:00:00+00:00      NaN
1986-03-14 00:00:00+00:00    0.035714
1986-03-17 00:00:00+00:00    0.017241
1986-03-18 00:00:00+00:00   -0.025424
1986-03-19 00:00:00+00:00   -0.017391
...
2021-02-24 00:00:00+00:00    0.005487
2021-02-25 00:00:00+00:00   -0.023705
2021-02-26 00:00:00+00:00    0.014804
2021-03-01 00:00:00+00:00    0.019623
2021-03-02 00:00:00+00:00   -0.012957
Name: adjClose, Length: 8814, dtype: float64
```

Compound these returns:

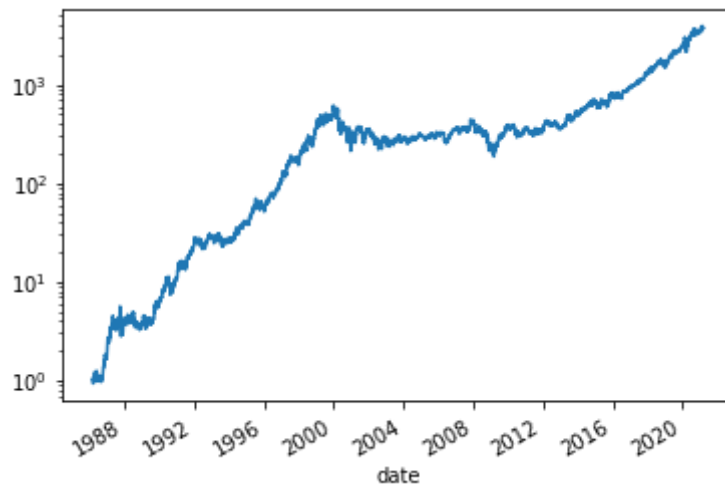
```
In [26]: # Example:
         r1 = 0.1
         r2 = 0.2

         # Compound return (includes "interest on interest"):
         (1+r1) * (1+r2) - 1
```

```
Out[26]: 0.32000000000000006
```

```
In [27]: r.add(1).cumprod().plot(logy=True)
```

```
Out[27]: <AxesSubplot:xlabel='date'>
```



Interpretation of this graph: this is the dollar amount in your account if you initially invest \$1 and then always reinvest all dividends.

Which part of this return comes from reinvesting dividends?

```
In [28]: # Example
dividend = 5
price    = 200

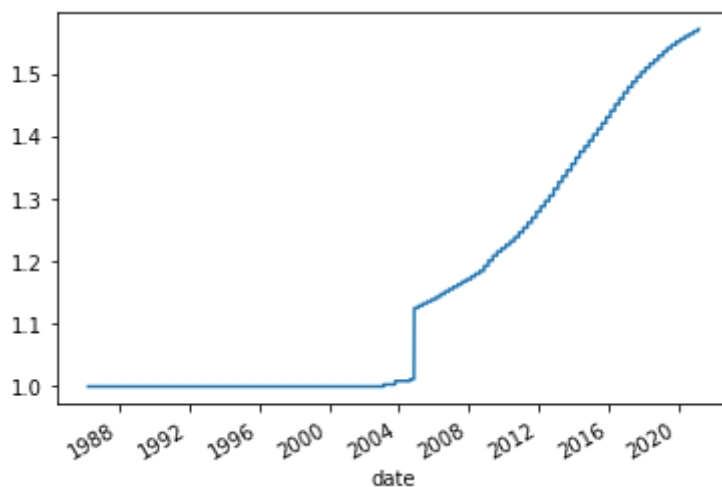
# How many new shares can we buy for this dividend:
dividend/price
```

Out[28]: 0.025

Compound dividend growth:

```
In [64]: (data.divCash / data.close).add(1).cumprod().plot()
```

Out[64]: <AxesSubplot:xlabel='date'>



Interpretation of this graph: this is the number of shares we have in our account if we start with 1 share and then reinvest all dividends (ignoring stock splits).