8.2 Historical Stock Data

In [1]:

```
data_dir = 'data/stocks'

djia = ['AXP','BA','CAT','CSCO','CVX','DD','DIS','GE', \
        'GS','HD','IBM','INTC','JNJ','JPM','KO','MCD', \
        'MMM','MRK','MSFT','NKE','PFE','PG','T','TRV', \
        'UNH','UTX','V','VZ','WMT','XOM']

favs = ['AAPL']

stocks = djia + favs

import os
os.makedirs(data_dir, exist_ok=True)
```

Alpha Vantage

The following cells retrieve a history of daily trading data for a specified set of stock ticker symbols. These functions use the free Alpha Vantage data service.

The service requires an personal api key which can be claimed here in just a few seconds. Place the key as a string in a file `data/api_key.txt` in the data directory as this notebook (note: api_key.txt is not distributed with the github repository). The function `api_key()` returns the key stored in `api_key.txt`.

In [2]:

```
def api_key():
    "Read api_key.txt and return api_key"
    try:
        with open('data/api_key.txt') as fp:
            line = fp.readline()
    except:
        raise RuntimeError('Error while attempting to read data/api_key.txt')
    return line.strip()
```

The function `alphavantage(s)` returns a pandas dataframe holding historical trading data for a stocker ticker symbol specified by `s`.

In [3]:

```
import os
import requests
import pandas as pd

def alphavantage(symbol=None):
    if symbol is None:
        raise ValueError("No symbol has been provided")
    payload = {
        "function": "TIME_SERIES_DAILY_ADJUSTED",
        "symbol": symbol,
        "outputsize": "full",
        "datatype": "json",
        "apikey": api_key(),
    }
    api_url = "https://www.alphavantage.co/query"
    try:
        response = requests.get(api_url, params=payload)
    except:
        raise ValueError("No response using api key: " + api_key)
    data = response.json()
    k = list(data.keys())
    metadata = data[k[0]]
    timeseries = data[k[1]]
    S = pd.DataFrame.from_dict(timeseries).T
    S = S.apply(pd.to_numeric)
    S.columns = [h.lstrip('12345678. ') for h in S.columns]
    return S

alphavantage('AAPL').head()
```

Out[3]:

	open	high	low	close	adjusted close	volume	dividend amount	split coefficient
2000-01-03	104.87	112.50	101.69	111.94	3.5554	4783900	0.0	1.0
2000-01-04	108.25	110.62	101.19	102.50	3.2556	4574800	0.0	1.0
2000-01-05	103.75	110.56	103.00	104.00	3.3032	6949300	0.0	1.0
2000-01-06	106.12	107.00	95.00	95.00	3.0174	6856900	0.0	1.0
2000-01-07	96.50	101.00	95.50	99.50	3.1603	4113700	0.0	1.0

`get_stock_data(symbols)` retrieves trading data for a list of symbols and stores each in seperate file in the data directory. The file name is the ticker symbol with a `.csv` suffix.

In [4]:

```
def get_stock_data(symbols, service=alphavantage):
    if isinstance(symbols, str):
        symbols = [symbols]
    assert all(isinstance(s, str) for s in symbols)
    for s in symbols:
        print('downloading', s, end='')
        k = 3
        while k > 0:
            try:
                k -= 1
                S = service(s)
                S.to_csv(os.path.join(data_dir, s + '.csv'))
                print(' success')
                break
            except:
                print(' fail', end='')
            print('')

get_stock_data(['AAPL'])
```

```
downloading AAPL success
```

Download Selected Ticker Symbols

In [5]:

```
get_stock_data(stocks)
```

```
downloading AXP success
downloading BA success
downloading CAT success
downloading CSCO success
downloading CVX success
downloading DD fail
 fail
 fail
downloading DIS success
downloading GE success
downloading GS success
downloading HD success
downloading IBM success
downloading INTC success
downloading JNJ success
downloading JPM success
downloading KO success
downloading MCD success
downloading MMM success
downloading MRK success
downloading MSFT success
downloading NKE success
downloading PFE success
downloading PG success
downloading T success
downloading TRV success
downloading UNH success
downloading UTX success
downloading V success
downloading VZ success
downloading WMT success
downloading XOM success
downloading AAPL success
```

```
In [ ]:
```