Create CSVs for top 10 stocks

In [2]:

```
# Load in packages
from pandas_datareader import data, wb
from datetime import datetime
import numpy as np
import pandas as pd
from scipy import stats
from itertools import chain
import matplotlib.pyplot as plt
import operator
import sys

import math, time
import itertools
from sklearn import preprocessing
from datetime import datetime
from operator import itemgetter
from sklearn.metrics import mean_squared_error
from math import sqrt
import tensorflow as tf
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation
from keras.layers.recurrent import LSTM


%matplotlib inline
```

```
Using TensorFlow backend.
```

Get data

In [3]:

```
month_dict = {'01': 'Jan', '02': 'Feb', '03': 'Mar', '04': 'Apr', '05': 'May', '06': 'Jun',
              '07': 'Jul', '08': 'Aug', '09': 'Sep', '10': 'Oct', '11': 'Nov', '12': 'Dec'}

def get_stock_data(ticker, start, end):
    """
    Input: stock ticker, start date, end date
    Output: numpy array of stock data
    """

    # Function to check if date is valid
    def check_date(date):
        return len(date) == 3 and len(date[0]) == 4 and len(date[1]) == 2 and len(date[2]) == 2

    # Format start and end days and check validity of dates
    start = start.split('-')
    end = end.split('-')
    errors = not (check_date(start) and check_date(end))
    if errors:
        print 'Your dates need to be of the form \'YYYY-MM-DD\'!'
        return
    start_date = month_dict[start[1]] + "+" + start[2] + "%2C+" + start[0]
    end_date = month_dict[end[1]] + "+" + end[2] + "%2C+" + end[0]

    start_dt = datetime(int(start[0]), int(start[1]), int(start[2]))
    end_dt = datetime(int(end[0]), int(end[1]), int(end[2]))
    timelapse = end_dt - start_dt
    business_days = np.busday_count(start_dt, end_dt)
#     print business_days, "business days"

    url = "http://finance.google.com/finance/historical?q=" + ticker +
"&startdate=" + start_date + "&enddate=" + end_date +
"&num=30&ei=SxDHWfiBHMzYjAGLnrr4Ag&output=csv"
    col_names = ['Date','Open','High','Low','Close','Volume']
    data = pd.read_csv(url, header=0, names=col_names)
#     print data.shape
#     data = np.array(data)[::-1] # order rows in chronological order
#     data = data.astype(float)
    return data
```

In [ ]:

```

```

In [4]:

```
x = get_stock_data('MMM', '2006-01-01', '2017-11-01')
# x.to_csv('data/test.csv')
x.shape
```

Out[4]:

```
(2980, 6)
```

In [32]:

```
x.head()
```

Out[32]:

	Date	Open	High	Low	Close	Volume
0	3-Nov-17	174.00	174.26	171.12	172.50	59398631
1	2-Nov-17	166.60	168.50	165.28	168.11	41393373
2	1-Nov-17	169.87	169.94	165.61	166.89	33637762
3	31-Oct-17	167.90	169.65	166.94	169.04	36046828
4	30-Oct-17	163.89	168.07	163.72	166.72	44700772

Save data

In [8]:

```
stocks = ['GOOGL','INTC', 'AAPL', 'CSCO', 'AMD', 'QCOM', 'NVDA', 'AMZN', 'MS
FT', 'IBM']
start_date = '2006-01-01'
end_date = '2017-11-01'

for ticker in stocks:
    file_name = 'data/' + ticker + '_' + start_date + '_to_' + end_date + '.
csv'
    print file_name
    data = get_stock_data(ticker, start_date, end_date)
    print data.shape
    data.to_csv(file_name)
```

```
data/GOOGL_2006-01-01_to_2017-11-01.csv
(2979, 6)
data/INTC_2006-01-01_to_2017-11-01.csv
(2979, 6)
data/AAPL_2006-01-01_to_2017-11-01.csv
(2979, 6)
data/CSCO_2006-01-01_to_2017-11-01.csv
(2979, 6)
data/AMD_2006-01-01_to_2017-11-01.csv
(2978, 6)
data/QCOM_2006-01-01_to_2017-11-01.csv
(2979, 6)
data/NVDA_2006-01-01_to_2017-11-01.csv
(2979, 6)
data/AMZN_2006-01-01_to_2017-11-01.csv
(2979, 6)
data/MSFT_2006-01-01_to_2017-11-01.csv
(2979, 6)
data/IBM_2006-01-01_to_2017-11-01.csv
(2980, 6)
```

In [37]:

```
len(stocks)
```

Out[37]:

```
32
```

In [ ]:

```
```