1 Python Folder:

Python is used for data scraping. You need to feed in a .csv file with ticker symbols and that file name must be changed in both files in order for the scraping to work

1.1 scrape.py

gets the data from yahoo finance in the format of month, date, year & price

```
from bs4 import BeautifulSoup
import requests
import os
import csv
import re

def parseTickers(file):
    tickers = []
    with open(file, 'rU') as csvfile:
        spamreader = csv.reader(csvfile, delimiter=',', quotechar=' '
        for row in spamreader:
            tickers.append(row[0])
    #ticker is an array that holds all symbols from TSX100
    return tickers


def scrapYahooPage(ticker):
    # craft url query
    stock = ticker
    # month, day, year
    start_month = '00'
    start_day = '1'
    start_year = '1960'
    end_month = '10'
    end_day = '10'
    end_year = '2014'
    interval = 'm'
    pg = 0
    lastdate = False
    # open CSV writer
    csvfile = open('output/scrape/'+ticker+'.csv', 'wb')
    writer = csv.writer(csvfile)
```

```
data_available = True
while data_available:
    url = 'https://ca.finance.yahoo.com/q/hp?s=' + stock + '&a='
    r = requests.get(url)
    soup = BeautifulSoup(r.text)
    data = soup.find(attrs={'class': 'yfnc_datamodoutline1'})
    if data:
        try:
            rows = data.findAll('td')
        except Exception:
            if(lastdate):
                print "Last row of prices: " + str(lastdate)
            else:
                print "No data available"
            data_available = False
            pass
        dateitem = []
        priceitem = []
        lastdate = " "
        lastmon = " "
        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'}
        # filter out rows for page end, dividend, split
        exclude = ['Close', 'Dividend', ":", "Split"]
        for row in rows:
            box = row.findAll(text=True)
            item = ','.join(box)
            if re.search('[a-zA-Z]+', item) and not any(n in item
                # date format 1 Mon day Year

                date = item.replace(",", "")
                date = date.replace('"', "")
                if lastmon not in date:
                    dateitem.append(date)
                else:
                    dateitem = [date]

            elif '-' in item:
                # date in format 2 year-mon-day
                datetxt = item.replace("-", " ")
                date = str(month_dict.get(str(datetxt[5:7]))) + "
                if lastmon not in date:
                    dateitem.append(date)
                else:
                    dateitem = [date]

            elif '.' in item and 'Dividend' not in item:
                # price object
                if dateitem: # add if date has already been set
                    priceitem.append(item)

            elif 'Dividend' in item:
                # Dividend payment, delete previous date
                dateitem = []
                priceitem = []


            if dateitem and priceitem:
                writer.writerow([dateitem[0], priceitem[0]])
                lastdate = dateitem[0]
                lastmon = lastdate[:3]
                dateitem = []
                priceitem = []

        pg += 66
        if pg > 594:
            data_available = False
    else:
        data_available = False
```

```
# import ticker symbols csv file
filename = 'input/missingSP500.csv'
offset = int(raw_input("Enter an offset to start: "))
index = 0

# loop over each ticker and get data
tickers = parseTickers(filename)
print "Fetching from file: " + filename + "..."


for ticker in tickers:

    # for .csv in ticker symbol
    ticker = ticker.replace('.csv', '')
    # for wikipedia stock names
    ticker = ticker.replace('.', '-')

    # now at EQR, 170
    if index >= offset:
        print "Scraping Stock ("+str(index)+"): " + ticker
        scrapYahooPage(ticker)
    index += 1
```

1.2 delete.py

removes duplicates and ensures there is only 1 date for each month

```
from bs4 import BeautifulSoup
import requests
import os
import csv
import re

def parseTickers(file):
    tickers = []
    with open(file, 'rU') as csvfile:
        spamreader = csv.reader(csvfile, delimiter=',', quotechar=' '
        for row in spamreader:
            tickers.append(row[0])
    #ticker is an array that holds all symbols from TSX100
    return tickers
```

```
def deleteDuplicates(ticker):
    rows = []
    lastmonth = " "
    with open('output/scrape/'+ticker+'.csv', 'rb') as inputfile:
        reader = csv.reader(inputfile)
        for row in reader:
            key = (row[0], row[1][:5])
            month = key[0][:3]
            if month == lastmonth:
                # 2 consequtive months are the same
                removed = rows.pop()
                print "Removed: " + str(removed)
            rows.append(row)
            lastmonth = month

    with open('output/delete/'+ticker+'.csv', 'wb') as outputfile:
        writer = csv.writer(outputfile)
        for row in rows:
            writer.writerow([row[0], row[1]])



# import ticker symbols csv file
filename = 'input/missingSP500.csv'
offset = int(raw_input("Enter an offset to start: "))
index = 0

# loop over each ticker and get data
tickers = parseTickers(filename)
print "Fetching from file: " + filename + "..."


for ticker in tickers:

    # for .csv in ticker symbol
    ticker = ticker.replace('.csv', '')
    # for wikipedia stock names
    ticker = ticker.replace('.', '-')
    print "Processing Stock ("+str(index)+"): " + ticker
    deleteDuplicates(ticker)
    index +=1
# output to a csv
```