STOCK MARKET DATA SCRAPING

May 1st, 2017

INTRO

Web scraping is one of the most architectural and structural necessities of any data project. It contributes time, resource and cost savings due to its efficient and accurate ability to use publicly available or social data to drive analysis and drive decisions.

In this project my goal was to scrape data from money.cnn.com/data/markets using Python’s Beautiful Soup library and export this data to an excel sheet for further analysis.

My project focusses on only part one which is data scraping and extraction. With this data, an analyst could run the script daily to gather numerical and qualitative data about stock markets, world markets, related news articles and economic indicators to drive investing decisions.

PYTHON SCRIPT

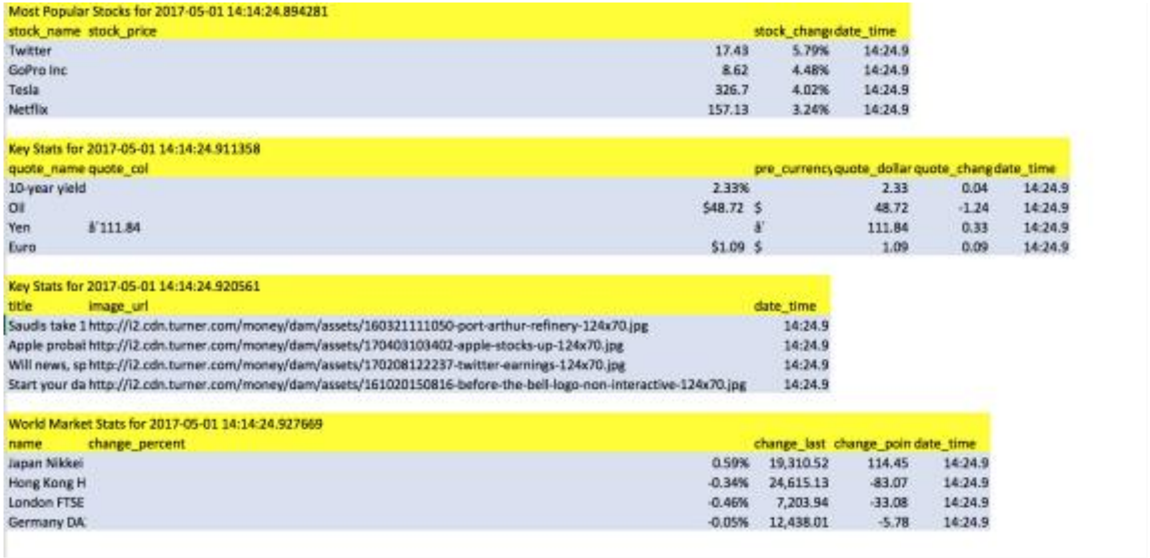
```
Created on Mon May  1 11:26:15 2017

@author: neeharikakaja
"""

# import libraries
import csv
from datetime import datetime
import urllib.request
from bs4 import BeautifulSoup
link = 'http://money.cnn.com/data/markets/'
# specify the url
with urllib.request.urlopen(link) as url:
  page = url.read()
# parse the html using beautiful soap and store in variable `soup`
soup = BeautifulSoup(page, 'html.parser')



#getting and writing most popular stocks
most_pop_stocks = soup.find('ul', attrs={'class': 'module-body wsod most-popular-stocks'})
all_li = (most_pop_stocks.findAll('li'))

#prep csv
with open('index.csv', 'a') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['Most Popular Stocks for ' + str(datetime.now())])
    writer.writerow(['stock_name', 'stock_price', 'stock_change', 'date_time'])




list = (0,1,2,3)
for i in (list):
    stock_name = (all_li[i].find('span', attrs={'class': 'column stock-name'})).text
    stock_price = (all_li[i].find('span', attrs={'class': 'column stock-price'})).text
    stock_change = (all_li[i].find('span', attrs={'class': 'column stock-change'})).text
#    print(stock_name + " " + stock_price + " " + stock_change)
    with open('index.csv', 'a') as csv_file:
        writer = csv.writer(csv_file)
        writer.writerow([stock_name, stock_price, stock_change, datetime.now()])

#finish with blank line for spacing
with open('index.csv', 'a') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['\n'])
```

```
#getting and writing key stats
most_pop_stocks = soup.find('ul', attrs={'class': 'module-body wsod key-stats'})
all_li = (most_pop_stocks.findAll('li'))

#prep csv
with open('index.csv', 'a') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['Key Stats for ' + str(datetime.now())])
    writer.writerow(['quote_name', 'quote_col', 'pre_currency_symbol', 'quote_dollar', 'quote_change', 'date_time'])


list = (0,1,2,3)
for i in (list):
    quote_name = (all_li[i].find('span', attrs={'class': 'column quote-name'})).text
    quote_col = (all_li[i].find('span', attrs={'class': 'column quote-col'})).text
    pre_currency_symbol = (all_li[i].find('span', attrs={'class': 'pre-currency-symbol'})).text
    quote_dollar = (all_li[i].find('span', attrs={'class': 'quote-dollar'})).text
    quote_change = (all_li[i].find('span', attrs={'class': 'column quote-change'})).text


#    print(stock_name + " " + stock_price + " " + stock_change)
    with open('index.csv', 'a') as csv_file:
        writer = csv.writer(csv_file)
        writer.writerow([quote_name, quote_col, pre_currency_symbol, quote_dollar, quote_change, datetime.now()])

#finish with blank line for spacing
with open('index.csv', 'a') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['\n'])
```

```
#getting and writing key stats
most_pop_stocks = soup.find('ul', attrs={'class': 'summary-list summary-list-thumbs'})
all_li = (most_pop_stocks.findAll('li'))

#prep csv
with open('index.csv', 'a') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['Key Stats for ' + str(datetime.now())])
    writer.writerow(['title', 'image_url', 'date_time'])


list = (0,1,2,3)
for i in (list):
    title = (all_li[i].find('figcaption', attrs={'class': 'thumb-caption'})).text
    image_url = (all_li[i].find('img')['src'])
    #    print(stock_name + " " + stock_price + " " + stock_change)
    with open('index.csv', 'a') as csv_file:
        writer = csv.writer(csv_file)
        writer.writerow([title, image_url, datetime.now()])

#finish with blank line for spacing
with open('index.csv', 'a') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['\n'])
```

```
#getting and writing world market stats
most_pop_stocks = soup.find('div', attrs={'class': 'module-body wsod world-markets'})
all_li = (most_pop_stocks.findAll('a'))

#prep csv
with open('index.csv', 'a') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['World Market Stats for ' + str(datetime.now())])
    writer.writerow(['name', 'change_percent', 'change_last', 'change_points', 'date_time'])


list = (0,1,2,3)
for i in (list):
    world_market_name = (all_li[i].find('h3', attrs={'class': 'world-market-name'})).text
    change_percent = (all_li[i].find('span', attrs={'class': 'world-market-change-percent'})).text
    change_last = (all_li[i].find('span', attrs={'class': 'world-market-change-last'})).text
    change_points = (all_li[i].find('span', attrs={'class': 'world-market-change'})).text

    with open('index.csv', 'a') as csv_file:
        writer = csv.writer(csv_file)
        writer.writerow([world_market_name, change_percent, change_last, change_points, datetime.now()])



#finish with blank line for spacing
with open('index.csv', 'a') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['\n'])
```

Results:



Most Popular Stocks for 2017-05-01 14:14:24.894281

stock_name	stock_price	stock_chang	date_time
Twitter	17.43	5.79%	14:24.9
GoPro Inc	8.62	4.48%	14:24.9
Tesla	326.7	4.02%	14:24.9
Netflix	157.13	3.24%	14:24.9

Key Stats for 2017-05-01 14:14:24.911358

quote_name	quote_col	pre_currency	quote_dollar	quote_chang	date_time
10-year yield	2.33%		2.33	0.04	14:24.9
Oil	$48.72	$	48.72	-1.24	14:24.9
Yen	å'111.84	å'	111.84	0.33	14:24.9
Euro	$1.09	$	1.09	0.09	14:24.9

Key Stats for 2017-05-01 14:14:24.920561

title	image_url	date_time
Saudis take 1	http://i2.cdn.turner.com/money/dam/assets/160321111050-port-arthur-refinery-124x70.jpg	14:24.9
Apple probal	http://i2.cdn.turner.com/money/dam/assets/170403103402-apple-stocks-up-124x70.jpg	14:24.9
Will news, sp	http://i2.cdn.turner.com/money/dam/assets/170208122237-twitter-earnings-124x70.jpg	14:24.9
Start your da	http://i2.cdn.turner.com/money/dam/assets/161020150816-before-the-bell-logo-non-interactive-124x70.jpg	14:24.9

World Market Stats for 2017-05-01 14:14:24.927669

name	change_percent	change_last	change_poin	date_time
Japan Nikkei	0.59%	19,310.52	114.45	14:24.9
Hong Kong H	-0.34%	24,615.13	-83.07	14:24.9
London FTSE	-0.46%	7,203.94	-33.08	14:24.9
Germany DA	-0.05%	12,438.01	-5.78	14:24.9

Implications:

My project focusses on only part one which is data scraping and extraction. With this data, an analyst could run the script daily to gather numerical and qualitative data about stock markets, world markets, related news articles and economic indicators to drive investing decisions.

I enjoyed working on this project greatly and I believe there is great use in the world of web scraping. In addition to the efficiency and economies of scale saved due to processing time, scraping permits flexibility as it allows one to alter their script as the web changes and adapt to new formats.

In the future I believe that this type of analysis could be replicated on larger data sets with reference to other major market websites such as Cnbc, Investopedia, Nasdaq and Wall Street Journal.