

# Data Updating

November 20, 2017

```
In [ ]: import urllib2
        from bs4 import BeautifulSoup as bs
        import pandas as pd
        import datetime
        import pickle
        import time
        with open('/Users/Lwmformula/Downloads/stockdata/sourcecode' +
                  '/datalist.pkl.txt', "r") as f:
            datalist = pickle.load(f)
        localpath = '/Users/Lwmformula/Downloads/stockdata/{ }.csv'
        def oscillator_update(df,t):
            MA_update(df,t)
            BB_update(df,t)
            EMA_update(df,t)
            MACD_DEM_OSC_update(df,t)
            RSI_SMA_create(df)
            KD_update(df,t)
            pdi_ndi_adx_update(df,t)
            obv_update(df,t)
            ROC_update(df,t)
            MFI_update(df,t)
            return df

        def get_historical_data(name, number_of_days):
            data = []
            url = "https://finance.yahoo.com/quote/" + name + "/history/"
            rows = bs(urllib2.urlopen(url).read(),"lxml")
            rows = rows.findAll('table')[1].tbody.findAll('tr')
            for each_row in rows:
                divs = each_row.findAll('td')
                try:
                    date_temp = datetime.datetime.strptime(divs[0].span.text, "%b %d, %Y")
                    date = date_temp.strftime("%-d/%-m/%Y")
                    data.append({'Date':date,
                                'Open':float(divs[1].span.text.replace(',','')),
                                'High':float(divs[2].span.text.replace(',','')),
                                'Low':float(divs[3].span.text.replace(',','')),
```

```

        'Adj Close':float(divs[5].span.text.replace(',','')),
        'Volume':float(divs[6].span.text.replace(',',''))
    })

    except:
        continue
    return data[:number_of_days]

def time_error(timetest,update_list):
    timetest_str = datetime.datetime.fromtimestamp(timetest)
    timetest_str = timetest_str.strftime("%d/%m/%Y")
    try:
        index = [i for i,_ in enumerate(update_list) if _['Date'] == timetest_str][0]
        index += 1
        signal = 1
        return signal,index
    except:
        signal = 0
        index = 99999
        return signal,index

for main in datalist:
    update_list = get_historical_data(main, 50)
    df = pd.read_csv(localpath.format(main),index_col=0)
    oldtime = df.index[-1]
    try:
        index = [i for i,_ in enumerate(update_list) if _['Date'] == oldtime][0]
    except:
        timetest = time.mktime(datetime.datetime.strptime(oldtime, "%d/%m/%Y"))
        timetest = timetest.timetuple()
        while True:
            signal, index = time_error(timetest,update_list)
            timetest += 86400
            if signal == 1:
                break
    if index == 0:
        print (main + ': No need to update!')
        continue
    reordered = [c for c in df.columns]
    update_list = update_list[:index]
    df2 = pd.DataFrame(update_list,columns=['Date','Open','High',
                                           'Low','Adj Close','Volume'])

    df2 = df2.set_index(['Date'])
    df2 = df2.iloc[::-1]
    df= pd.concat([df,df2])
    df = df[reordered]
    #oscillator_update(df,-index-1)
    df.to_csv(localpath.format(main),index=True)
    print (main, ':Updated!')

```