Using Python on a Raspberry Pi to gather Stock & Share Prices at regular intervals and Plot Graphs.

Introduction.

I like to track share prices of various shares and study graphs with moving averages on them. What I don't like is the time it takes to do this using a web-based package. Many brokers provide systems where you select each share in turn and then open a graph to your preferred time span and then select the moving averages you want shown. Even with the preferences saved it still takes a long time to review thirty or more different shares. Years ago my Dad had a DOS based programme called Microvest that was great for quickly reviewing lots of graphs. The problem with this programme was that each week you had to sit down with the Financial Times and type in each share price. What I wanted is the same fast and simple graphs with an arrangement to get the share prices automatically from the Internet. I decided to stick with the method Dad used and collect each share price once a week.

After a bit of searching I found the Ystockquote module that provides an easy method of collecting share price data from the Internet. I started with the intention of using a SQL database store the collected data. I found I could save data and use queries to extract data but could not get the data to automatically pass to Gnuplot. After a few days of very slow progress decided to keep things simple and go with a separate text file for each share. To pass data between a Python programme and Gnuplot I use Gnuplot.py. When you look at the programs you will find some bits that look like they were written by people who know what they are doing, that's the bits I have borrowed, the rest with its poor structure, inconsistent naming conventions and generally looking a bit crappy is all mine.

These programs are functional, I am using them every week to monitor my shares and others I am interested in.

The programs currently available are:

GetHistorical.py	Collects historical share price data for multiple companies.
GetRegular.py	Collects share price data each Friday for multiple companies.
PlotShares.py	Plots share prices data with rolling averages for all collected companies.
PlotSingle.py	Plots share prices data with rolling averages for one company.
ReOrder.py	Fixes data file issues, re-indexes, deletes duplicate records, etc...
SetWatch.py	Allows user to set a watch/bought/sold price on graphs.
AdjPrice.py	Allows adjustment of share price data.
52WeekHiLo.py	Collects daily the number of shares achieving 52 week Highs and Lows.
PlotHiLo.py	Plots the 52 week Highs and Lows.

For a description of each program and links to the installation documents see below.

Share data file names

The Share name (ticker reference) used by the Ystockquotes function can be found by using the Yahoo finance pages.		https://uk.finance.yahoo.com/

The ticker reference is used to create the file name for the data files.

Shares quoted on the London Stock Exchange will have .L at the end.

Rolls Royce			RR.L		will be saved in a file		RR.L.txt

Shares quoted on the New York Stock Exchange have no letter at the end.

Google (Alphabet Inc)		GOOG		will be saved in a file		GOOG.txt

The FSTE indexes are saved as follows.

FTSE 100			^FTSE		will be saved in a file		^FTSE.txt
FTSE 250			^FTMC		will be saved in a file		^FTMC.txt

Disclaimer

I will not be responsible for any monetary loss, loss of data, corruption of files or any other problems resulting from these programs or the installation processes.

For a description of each program and links to the installation documents see below.
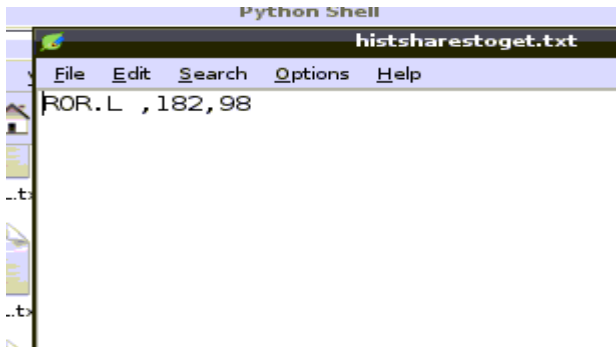
1.   **GetHistorical.py**

Installation Notes   http://www.rknorman.co.uk/GetHistorical.pdf
Program Listing   http://www.rknorman.co.uk/GetHistorical.py

Use this program to collect historical data. I like to have 2 years worth of data if possible.

   a.   Creates a temporary text file with the date of every Friday from a start date to Now
   b.   Collects the share price for all shares listed in a file called histsharestoget.txt.
   c.   The share price collected is the Closing price for each Friday.
   d.   The data is saved in a file using the ticker reference as the file name.
   e.   Collects share prices from an adjustable start date.
   f.   The share name is added to the sharestoget.txt file used for future scheduled weekly
        collections. See GetRegular
   g.   If the share data file already exists the new data is appended to it.
   h.   Each record has an incrementing index (pointer) used when plotting.
   i.   A watch/bought/sold price will be added to each record. See plotting below.

The histsharestoget.txt lists the shares that are to be collected. In the example above the GetHistorical program will collect shares prices from the company Rotork plc and store them in a file called ROR.L.txt in the SharesData directory.

The 98 at the end of the line is the index number that will be used for the first record collected. The next record will be 99 and so on.   When collecting for the first time set this value to 1.

A green line on the graphs is adjustable. Referred to as the watch/bought/sold line. I use it to initially show the value of the share when I first become interested in it. Later if I buy the share I update it to the price paid and then update again when the share is sold. See plot below.

The 182 value represents the watch/bought/sold price for the for the green line plot. It can be adjusted later by using SetWatch.py

If multiple records for the same date for the same share have been collected by mistake use the ReOrder program to delete the duplicate records and recreate the index.

_____

2.   **PlotShares.py**

Installation Notes   http://www.rknorman.co.uk/PlotShares.pdf
Program Listing   http://www.rknorman.co.uk/PlotShares.py

This program plots the shares graph using Gnuplot.   A graph for each share will open automatically. The plot window has to be closed before the next graph can be viewed. Use ReOrder to fix index issues if the plot is not as expected.

- a. Creates graph for each share listed in sharestoget.txt
- b. Each graph has four lines. Price, 8 week average, 16 week average & watch/bought/sold price.
- c. To create the four lines, a temporary CSV file is created for each line with its own index.
- d. gnuplot.py passes the CSV files to Gnuplot
- e. The plot window is a fixed size and position.
- f. Each plot window has to be closed before the next can be opened.
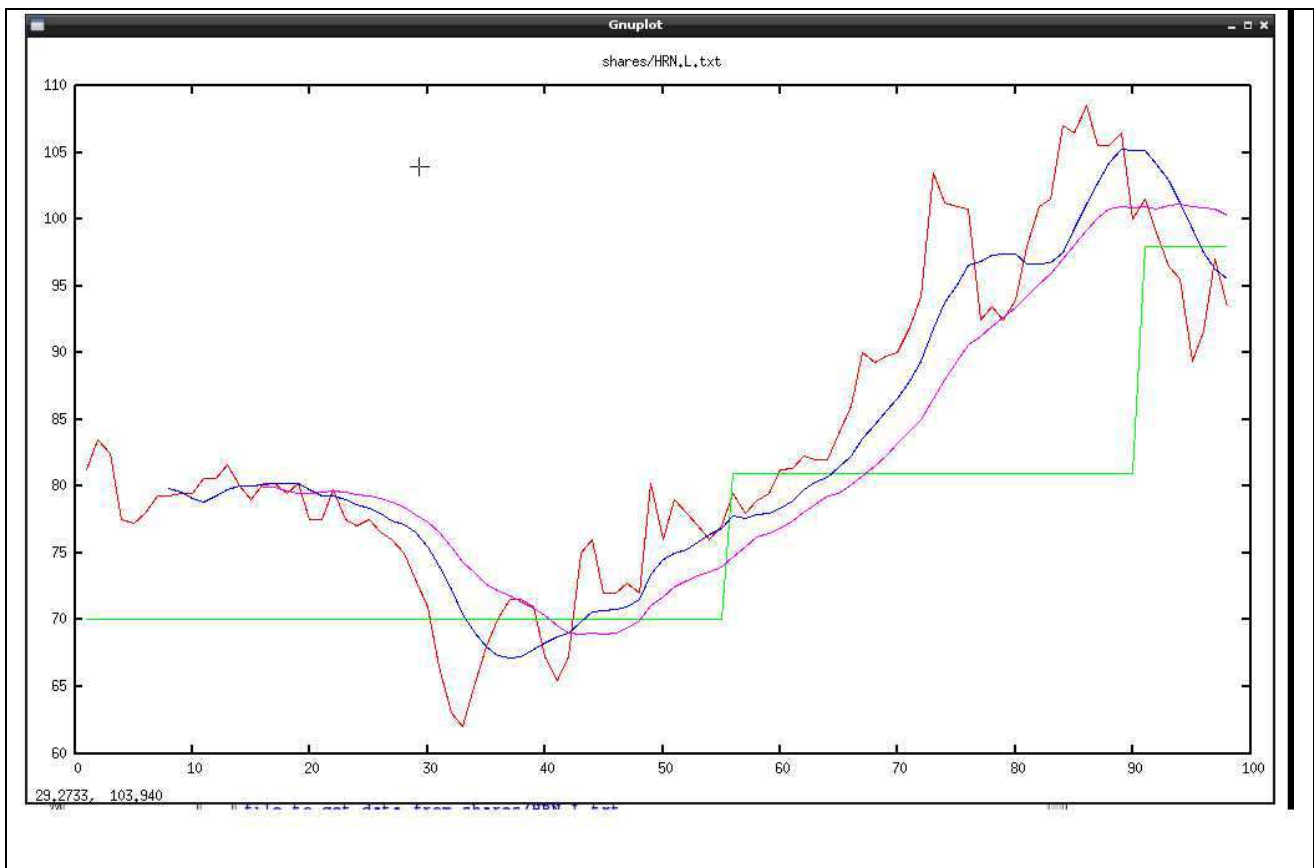
The actual plots produced are much clearer than the picture above suggests. The plot scaling is automatic.   If your green line (watch/bought/sold) price is significantly smaller or larger than the actual share price the graph view will be automatically zoomed out and not easy to see.

Red line shows the share price.
Blue line shows the 8 week average line.
Purple line shows the 16 week average line.
Green line shows the watch/bought/sold line.   This line is adjustable via SetWatch.py
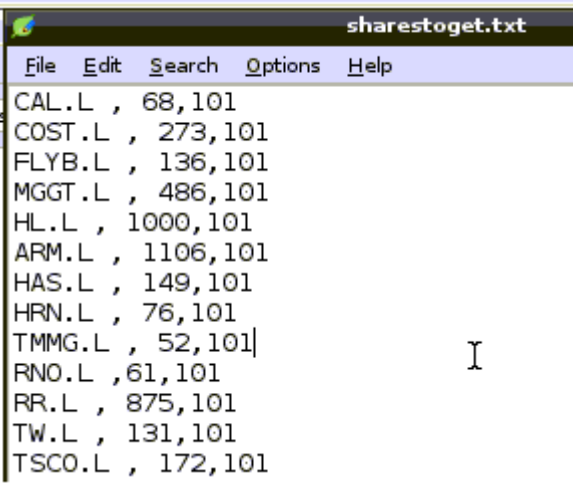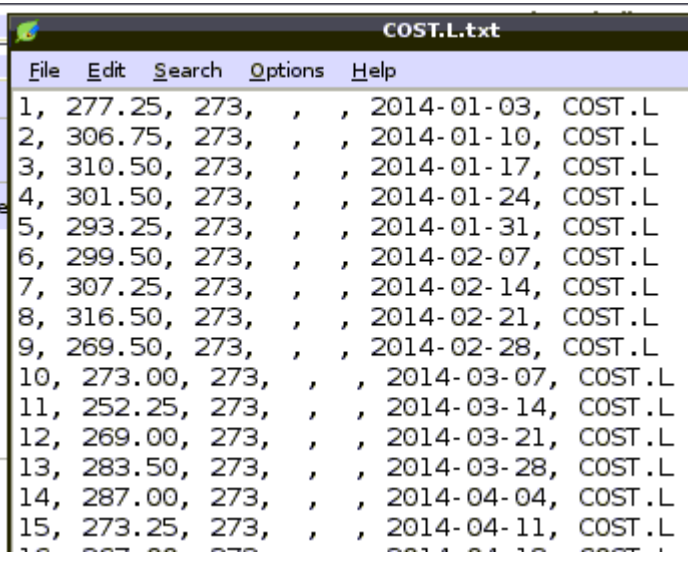
3.	**GetRegular.py**

Installation Notes	http://www.rknorman.co.uk/GetRegular.pdf
Program Listing	http://www.rknorman.co.uk/GetRegular.py

Uses the Ystockquote function to collect the Share Closing price each Friday at 19:00hrs. For testing purposes change the schedule to collect shares every few minutes.

   a.   Program waits for the scheduled time to collect data.
   b.   Collects the share price for all shares listed in a file called getshareprice.txt
   c.   Appends the index number, price, watch/boght/sold price and date to the share file.

<table>
<tr><td>

```
         sharestoget.txt
 File  Edit  Search  Options  Help
CAL.L , 68,101
COST.L , 273,101
FLYB.L , 136,101
MGGT.L , 486,101
HL.L , 1000,101
ARM.L , 1106,101
HAS.L , 149,101
HRN.L , 76,101
TMMG.L , 52,101|
RNO.L ,61,101
RR.L , 875,101
TW.L , 131,101
TSCO.L , 172,101
```

</td><td>

The sharestoget.txt file lists the shares to be collected via the GetRegular.py program.   This same list is used for the Plot and Reorder programs.

The format of this data is the same as that used in the histsharestoget.txt. file.

Ticker ref, watch/bought/sold price, index.

</td></tr>
</table>

```
                 COST.L.txt
 File   Edit   Search   Options   Help
1,  277.25,  273,    ,    , 2014-01-03,  COST.L
2,  306.75,  273,    ,    , 2014-01-10,  COST.L
3,  310.50,  273,    ,    , 2014-01-17,  COST.L
4,  301.50,  273,    ,    , 2014-01-24,  COST.L
5,  293.25,  273,    ,    , 2014-01-31,  COST.L
6,  299.50,  273,    ,    , 2014-02-07,  COST.L
7,  307.25,  273,    ,    , 2014-02-14,  COST.L
8,  316.50,  273,    ,    , 2014-02-21,  COST.L
9,  269.50,  273,    ,    , 2014-02-28,  COST.L
10,  273.00,  273,   ,    , 2014-03-07,  COST.L
11,  252.25,  273,   ,    , 2014-03-14,  COST.L
12,  269.00,  273,   ,    , 2014-03-21,  COST.L
13,  283.50,  273,   ,    , 2014-03-28,  COST.L
14,  287.00,  273,   ,    , 2014-04-04,  COST.L
15,  273.25,  273,   ,    , 2014-04-11,  COST.L
```

Above is an example of the data collected by GetRegular.   This file shows records for the Costain company. The format for data is: Index, share price, w/b/s price, space, space, date, ticker ref

3.   **ReOrder.py**

Installation Notes   http://www.rknorman.co.uk/ReOrder.pdf
Program Listing   http://www.rknorman.co.uk/ReOrder.py

Use this program to reorder the index numbers for all shares files; particularly useful if you have stopped one of the programs partway through and then restarted the program creating multiple index or date records.

    a. Reorders the index and date records for all shares listed in sharetoget.txt
    b. Reorders the price records in ascending date with ascending consecutive index numbers.
    c. Deletes multiple date records.
    d. Identifies missing date records

4.   **PlotSingle.py**

Installation Notes   http://www.rknorman.co.uk/PlotSingle.pdf
Program Listing   http://www.rknorman.co.uk/PlotSingle.py

Use this program to plot a graph with rolling averages for a single share.

5.   **AdjPrice.py**

Installation Notes   http://www.rknorman.co.uk/AdjPrice.pdf
Program Listing   http://www.rknorman.co.uk/AdjPrice.py

Use this program to adjust the share price if a company has artificially adjusted the share price by issuing more shares.

    a. Changes all share price records for a share between a start and end date by a selectable fixed factor.

6.   **SetWatch.py**

Installation Notes   http://www.rknorman.co.uk/SetWatch.pdf
Program Listing   http://www.rknorman.co.uk/SetWatch.py

Use this program to adjust the watch/bought/sold price.

    a. Changes all watch/bought/sold records for a share from a start date to the latest record.

7.   **52WeekHiLo.py**

Installation Notes   http://www.rknorman.co.uk/52WeekHiLo.pdf
Program Listing   http://www.rknorman.co.uk/52WeekHiLo.py

Use this program to automatically collect the number of shares on the London Stock Exchange attaining their 52 week high or low price.

    a. Collects the number of 52 week high and lows for each working day. (Mon- Fri)
    b. Saves date in file called 52s.L.txt.

8.   **PlotHiLo.py**

Installation Notes   http://www.rknorman.co.uk/PlotHiLo.pdf
Program Listing   http://www.rknorman.co.uk/PlotHiLo.py

Use this program to plot a graph using data from the 52s.L.txt file.

Features I hope to introduce in the future.

1. Adjustable start date for plotting graphs.
2. Adjustable rolling average periods.
3. Dates along the bottom of the graphs. Currently the scale is in weeks.
4. Company names along the top of the graphs. Currently it only shows the ticker reference.
5. Fix GetHistorical so it can be run on a Friday before close of markets without failing.
6. Mod SetWatch so it updates the watch/bought/sold price in the sharestoget.txt file.