

User Guide

What the program does

This is a stock analysis program. It provides statistical analyses and graphs for you to track the performances of stocks. You can also build a portfolio to see its return over a period. This program only analyzes the closing prices of stocks, since closing price is an accurate indicator of a stock's daily performance.

How to run the program

The program extracts data from Yahoo! Finance, so you need to get Internet access before you run the program. You also need to install python2.7, its libraries NumPy, pandas, matplotlib, Tkinter and requests. The program is run under Linux. Please see the detailed installation guide below.

Step1:

- Connect to the Internet

Step2:

- Open the terminal

Step3:

- Check whether you have python2.7 by typing the following command in the terminal:
 - `$ python --version`
- If python2.7 is not installed, use this command to install python2.7:
 - `$ sudo apt-get install python2.7`

Step 4:

- Install Numpy, pandas, matplotlib using the following command:
 - `$ sudo apt-get install python-numpy python-matplotlib python-pandas`

Step 5:

- Install Tkinter using the following command:
 - `$ sudo apt-get install python-tk`

Step 6:

- Install requests using the following command:
 - `$ sudo easy_install requests`

Step 7:

- Using the following command to run the program:
 - `$ cd python_project`
 - `$ python -u Main.py`

Libraries used in the program

NumPy: Numpy is an extension package to python, adding support for large, multidimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays. Numpy arrays are a much more efficient way of storing and manipulating numerical data than the other built-in Python data structures.

Pandas: Pandas provides rich data structures and functions designed to make working with structured data fast, easy, and expressive. One of the most important objects in pandas is DataFrame, a two-dimensional tabular and column-oriented data structure with both row and

column labels. It provides sophisticated indexing functionality, making it easy to reshape, slice and dice, perform aggregations, and select subsets of data. For financial users, pandas features rich, high-performance time series functionality and tools well-suited for working with financial data.

Matplotlib: Matplotlib is the most popular Python library for producing plots and other 2D data visualizations. It provides a comfortable interactive environment for plotting and exploring the data. The plots are interactive; you can zoom in on a section of the plot and pan around the plot using the toolbar in the plot window.

Tkinter:

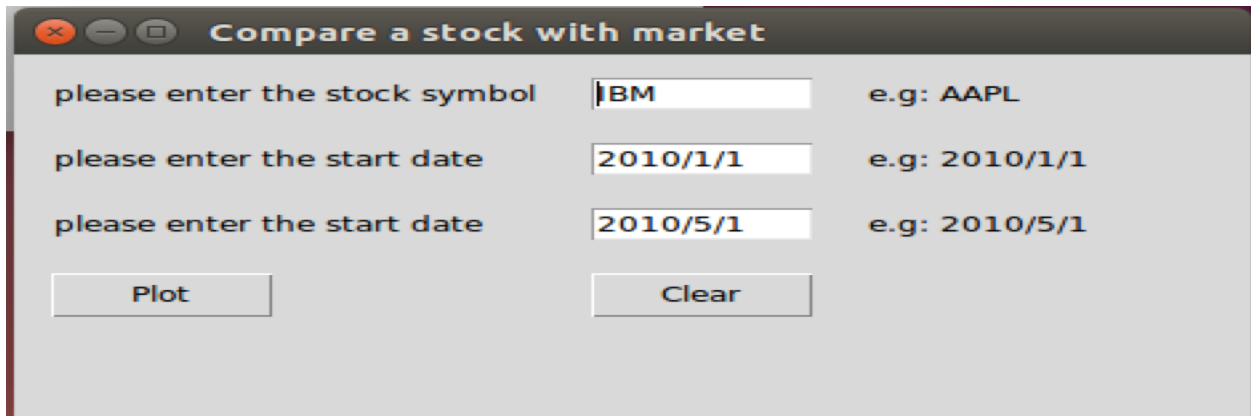
Tkinter is a GUI (graphical user interface) widget set for Python. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. widgets used in our project are:

- **Button:** The Button widget is used to display buttons in your application.
- **Entry:** The Entry widget is used to display a single-line text field for accepting values from a user.
- **Frame:** The Frame widget is used as a container widget to organize other widgets.
- **Label:** The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
- **LabelFrame:** A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.

- `tkMessageBox`: This module is used to display message boxes in your applications.

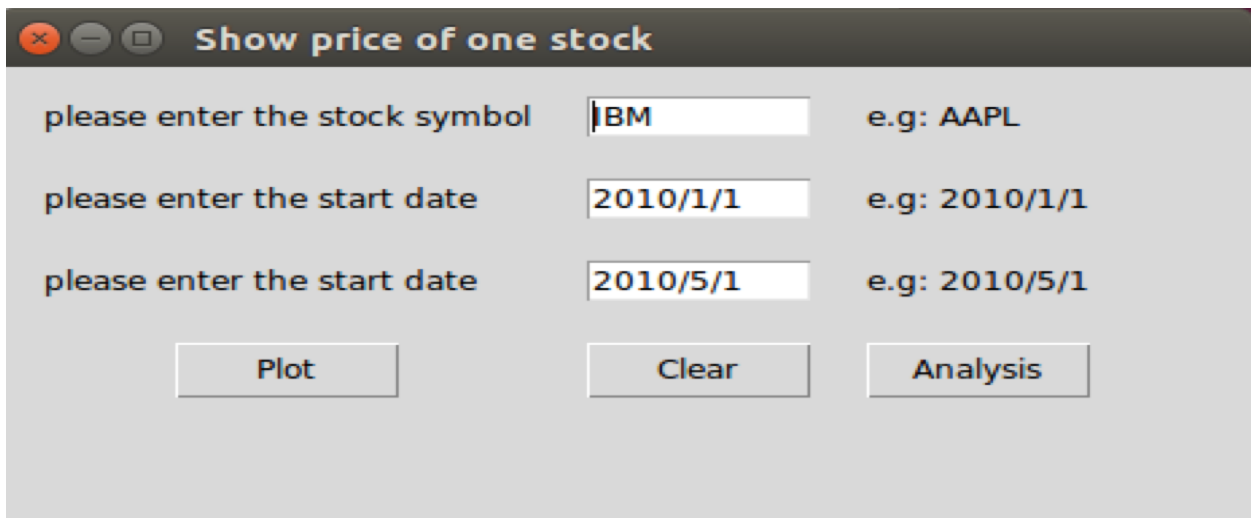
What inputs the program take and what outputs the program produce

After entering the command `$ python -u Main.py` in the terminal, it will create a window that consists of 6 buttons. This program has 4 main parts.

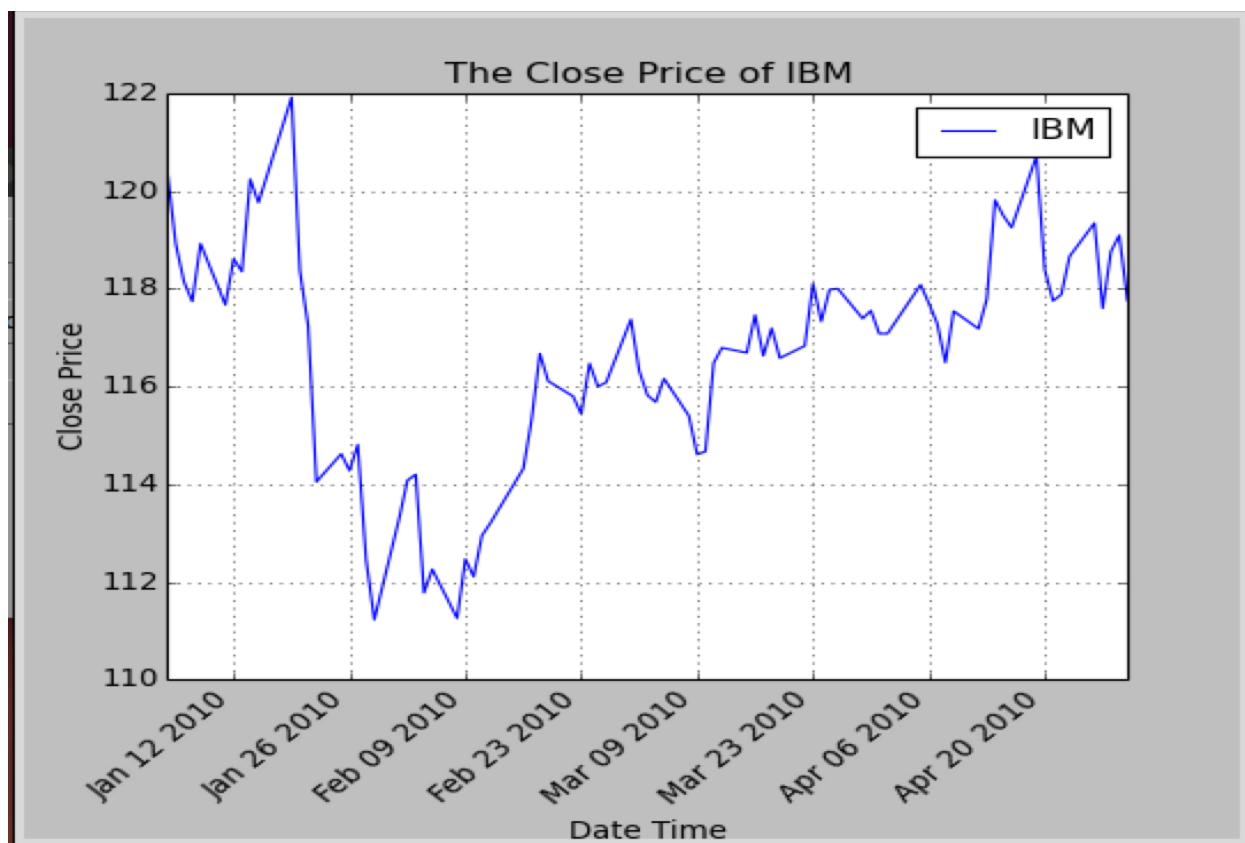


The screenshot shows a Tkinter window titled "Compare a stock with market". It has a light gray background and a dark gray title bar. Inside the window, there are three rows of input fields. The first row is labeled "please enter the stock symbol" and contains a text box with "IBM" and an example "e.g: AAPL". The second row is labeled "please enter the start date" and contains a text box with "2010/1/1" and an example "e.g: 2010/1/1". The third row is labeled "please enter the start date" (likely a typo for end date) and contains a text box with "2010/5/1" and an example "e.g: 2010/5/1". At the bottom, there are two buttons: "Plot" and "Clear".

The first part below takes a stock symbol and a trading period from the user. The user can then choose to see the stock's daily price in a line plot and its descriptive statistics (e.g., maximum, minimum and mean price) in a data frame over the period.



The screenshot shows a Tkinter window titled "Show price of one stock". It has a light gray background and a dark gray title bar. Inside the window, there are three rows of input fields. The first row is labeled "please enter the stock symbol" and contains a text box with "IBM" and an example "e.g: AAPL". The second row is labeled "please enter the start date" and contains a text box with "2010/1/1" and an example "e.g: 2010/1/1". The third row is labeled "please enter the start date" (likely a typo for end date) and contains a text box with "2010/5/1" and an example "e.g: 2010/5/1". At the bottom, there are three buttons: "Plot", "Clear", and "Analysis".



Show price of one stock

please enter the stock symbol e.g: AAPL

please enter the start date e.g: 2010/1/1

please enter the start date e.g: 2010/5/1

The Close Price Analysis of the Stock

count	82.000000
mean	116.649512
std	2.286720
min	111.230000
25%	115.410000
50%	117.130000
75%	118.060000
max	121.910000

Name: Adj Close, dtype: float64

The second part asks the user for a stock symbol and a trading period as well, after which it plots a graph comparing the stock's daily performance against the S&P 500 (stock market index) throughout the trading period. In the program, daily performance is calculated as the percentage change between the starting price and the daily price in the period.

Compare a stock with market

please enter the stock symbol

IBM

e.g: AAPL

please enter the start date

2010/1/1

e.g: 2010/1/1

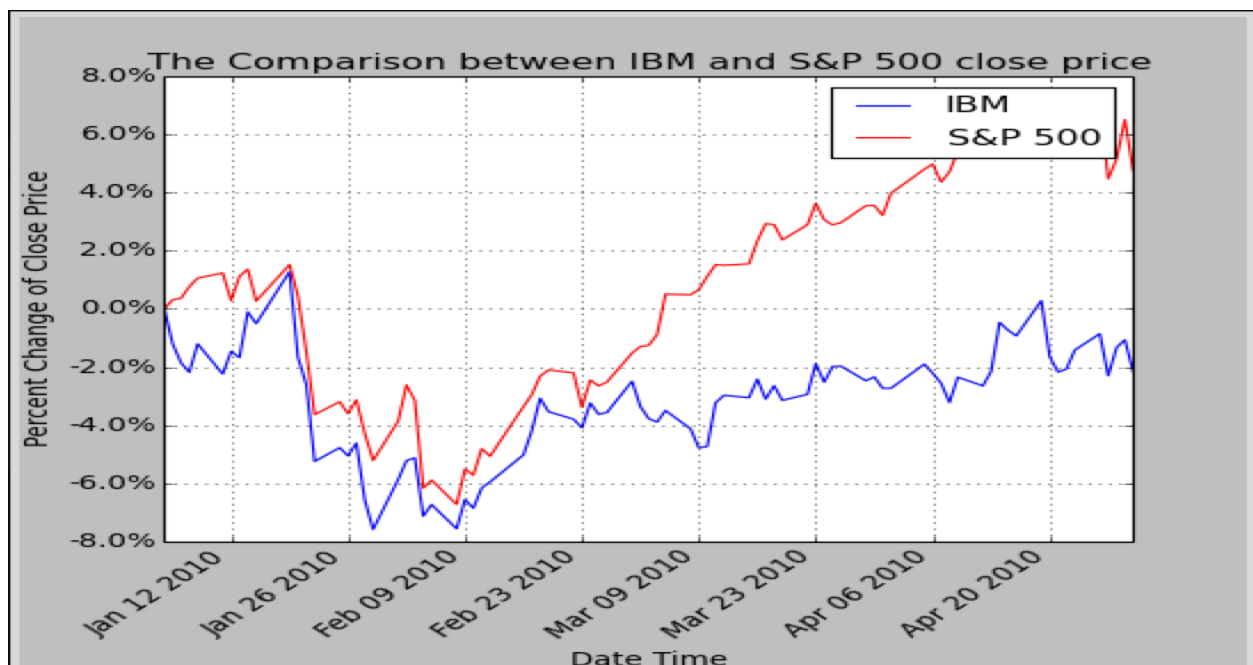
please enter the start date

2010/5/1

e.g: 2010/5/1

Plot

Clear



The third part allows the user to enter at most four stock symbols and date range. It can then plot the daily performances of the stocks in a line graph.

Compare prices of several stocks

please enter the first stock symbol

IBM

please enter the start date

2010/1/1

please enter the second stock symbol

AAPL

please enter the start date

2010/5/1

please enter the third stock symbol

JPM

date e.g: 2010/1/1

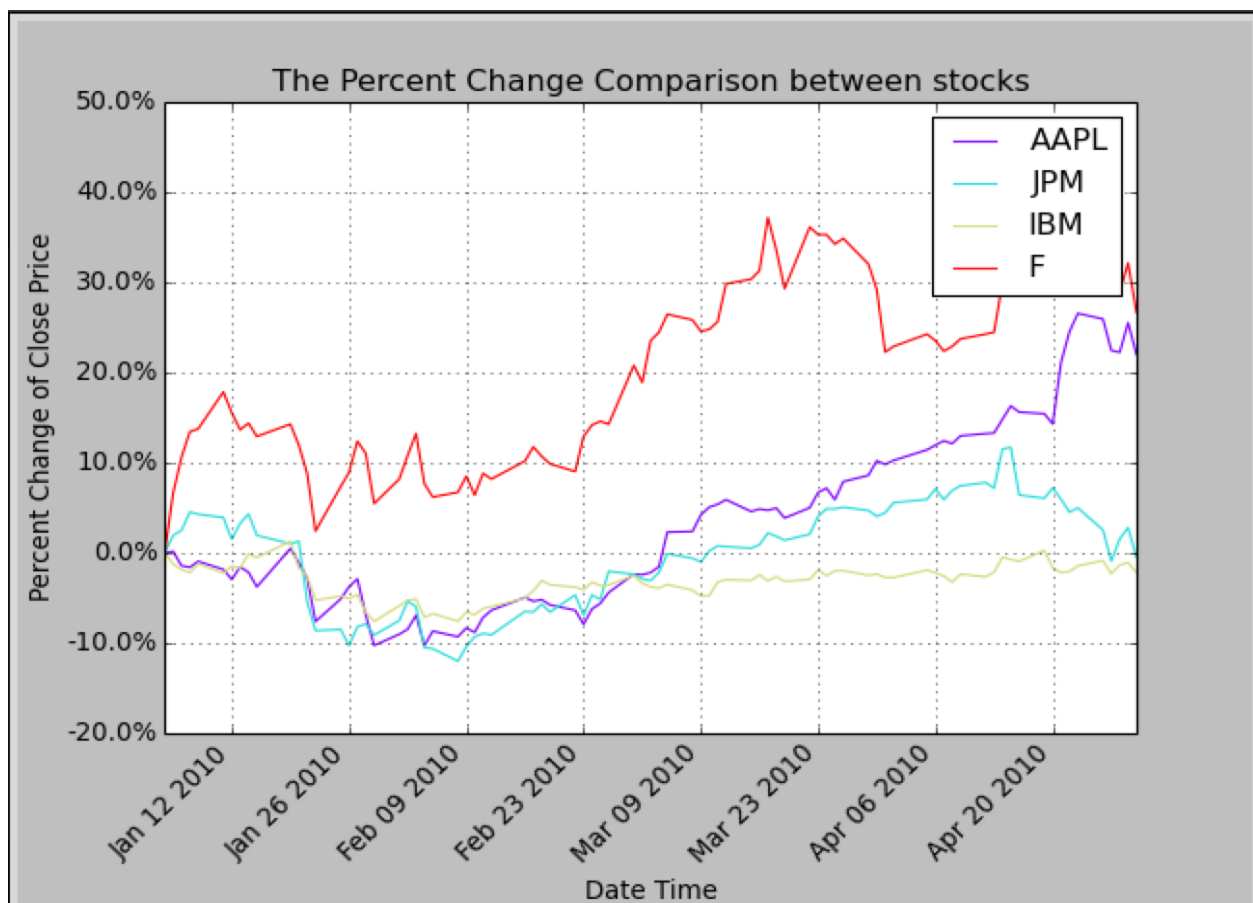
please enter the fourth stock symbol

F

stock symbol e.g: AAPL

Plot

Clear



The last part is a portfolio analysis, which inputs are a date range, at most four stock symbols and each stock's trading volume.

The screenshot shows a window titled "Portfolio analysis". It contains four rows of input fields for stock names and amounts, and two date range inputs. The first row has "BM" and "20". The second row has "AAPL" and "10". The third row has "JPM" and "20". The fourth row has "F" and "20". To the right of these are two date inputs: "2010/1/1" and "2010/5/1". There is a "Clear" button next to the first date. Below the date inputs are two dropdown menus: "statistics of the portfolio" and "portfolio performance". To the right of these are two buttons: "Analysis" and "Plot".

It has two analyses displayed in data frames. One analysis is the descriptive statistics of the portfolio (e.g., maximum and minimum of the portfolio price, start and end weight of each stock), the other one is the correlation among stock performances in the portfolio.

This is a close-up of the "Portfolio analysis" window. It shows the date range inputs: "please enter the start date" with "2010/1/1" and "please enter the start date" with "2010/5/1". Below these is a dropdown menu with three options: "correlation of the stocks changes", "statistics of the portfolio", and "portfolio performance". To the right of the dropdown are two buttons: "Analysis" and "Plot".

Statistics of the portfolio.

	AAPL	JPM	IBM	F	Portfolio
trading days	82.000000	82.000000	82.000000	82.000000	82.000000
mean	299.209756	766.990244	2332.990244	229.519512	3628.709756
std	28.891960	45.015914	45.734401	20.300748	124.152431
min	259.800000	676.400000	2224.600000	190.400000	3367.400000
25%	274.825000	727.250000	2308.200000	211.100000	3530.750000
50%	290.450000	776.000000	2342.600000	232.900000	3657.650000
75%	319.175000	803.250000	2361.200000	247.350000	3720.400000
max	366.400000	858.600000	2438.200000	268.000000	3840.100000
start weight	0.079187	0.210236	0.658497	0.052080	1.000000
end weight	0.095089	0.205901	0.634073	0.064936	1.000000
total return	0.220035	-0.004944	-0.021683	0.266807	0.016002

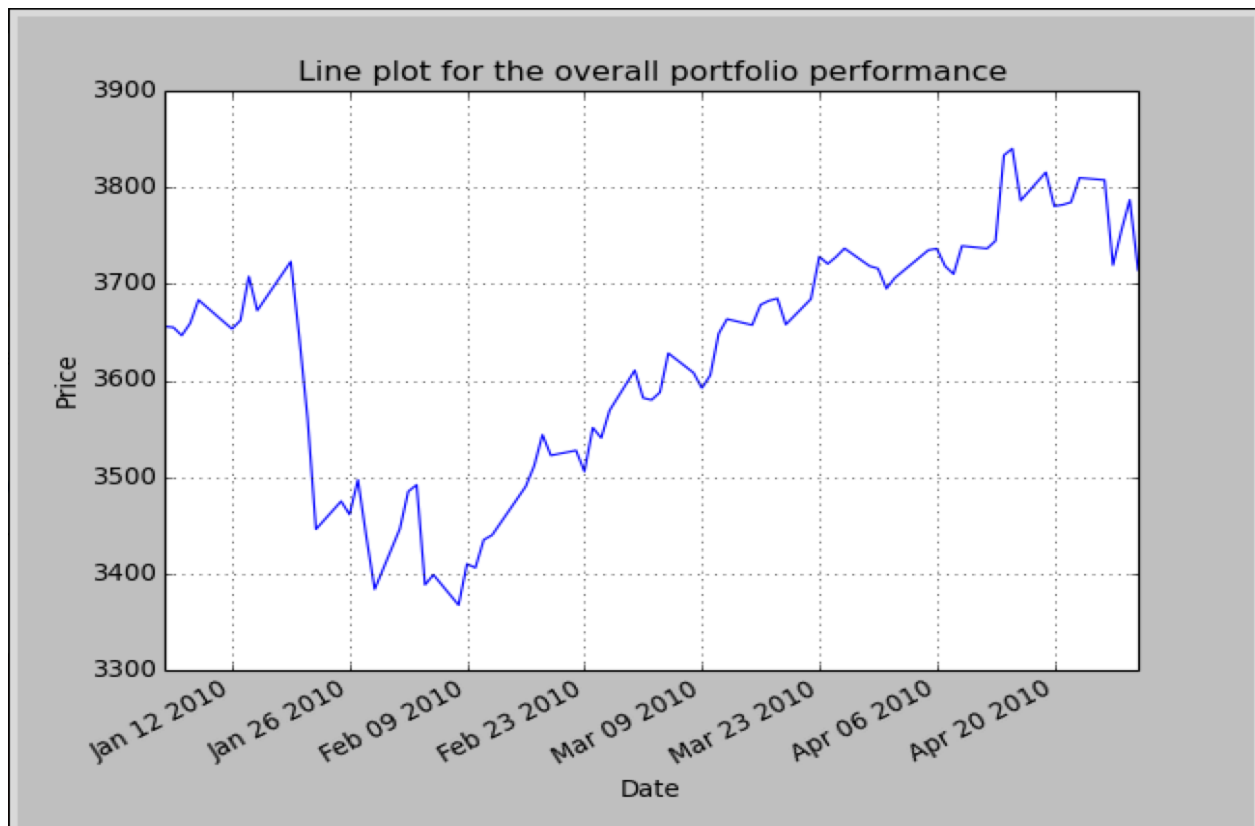
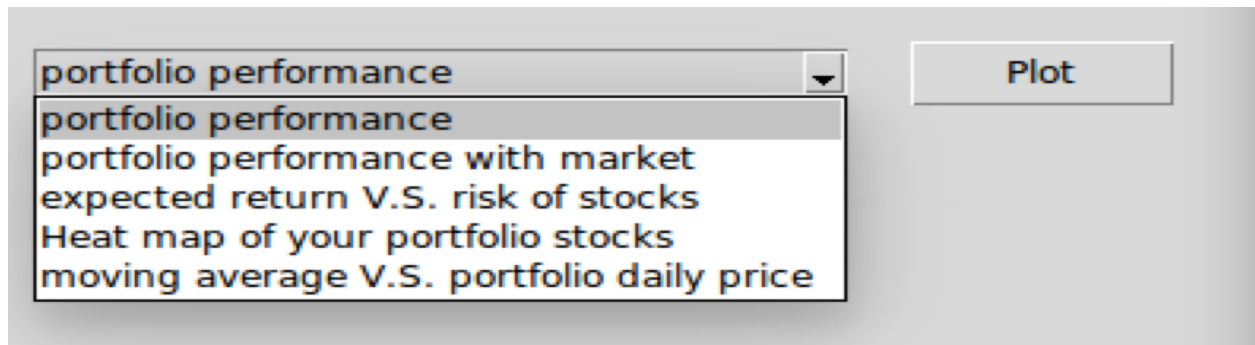
[11 rows x 5 columns]

Correlation of the stocks changes.

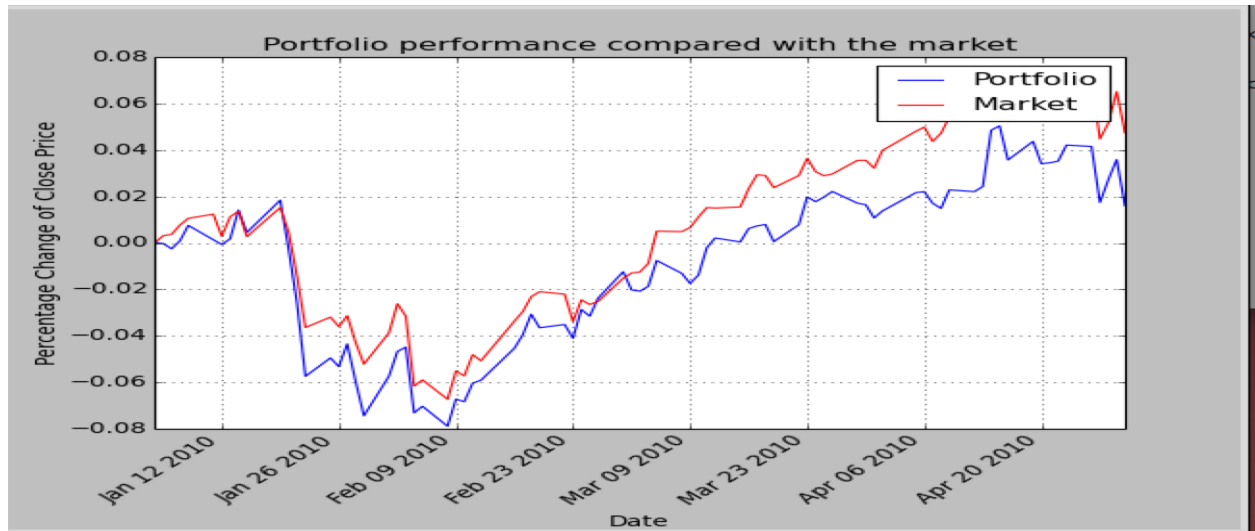
	AAPL	JPM	IBM	F
AAPL	1.000000	0.747880	0.612636	0.818090
JPM	0.747880	1.000000	0.806191	0.704992
IBM	0.612636	0.806191	1.000000	0.493712
F	0.818090	0.704992	0.493712	1.000000

[4 rows x 4 columns]

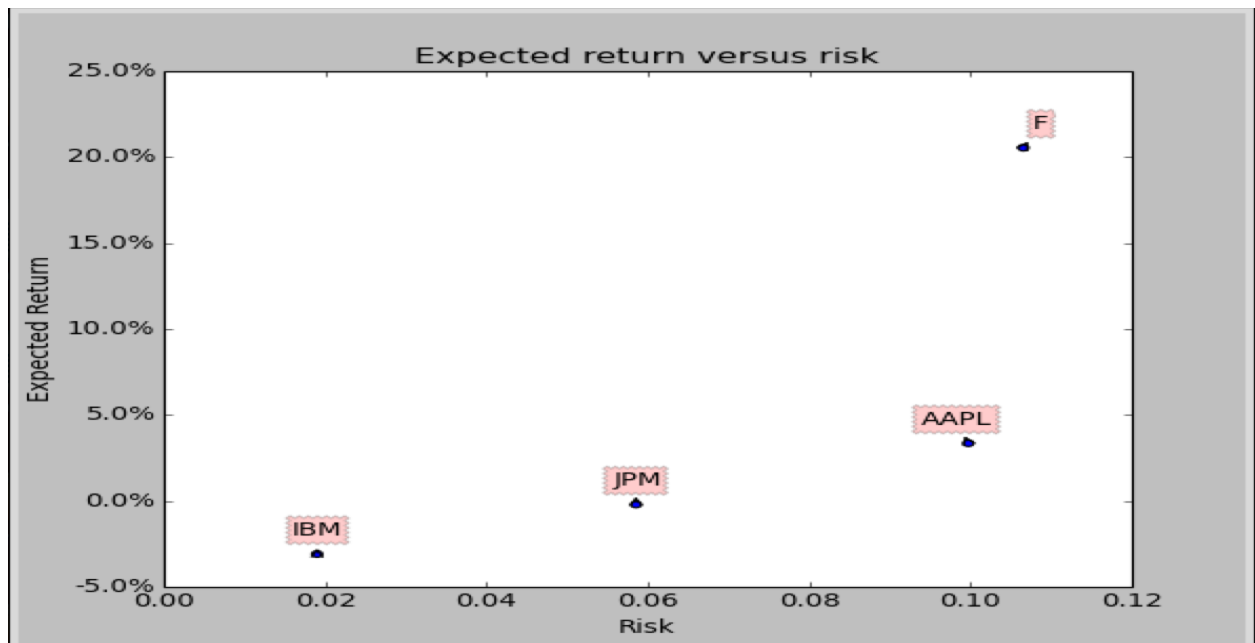
Besides the analyses, it has five plots for the user to choose from. The first one is a line plot showing the portfolio price during over the period.



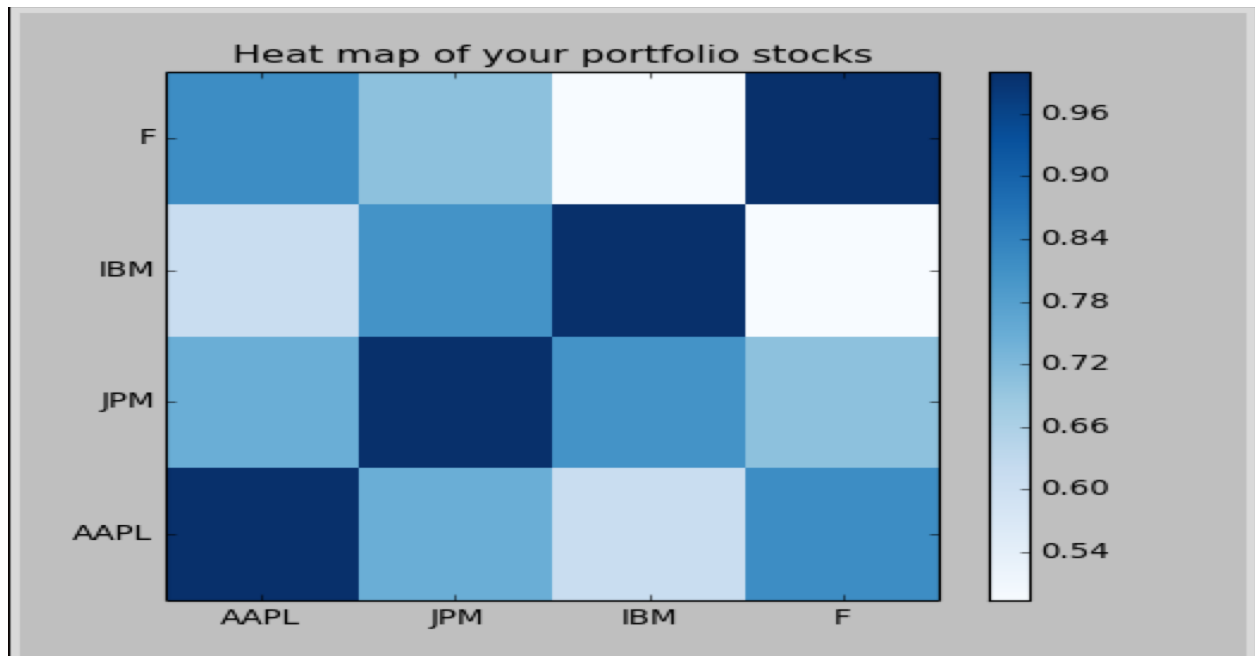
The second plot below is a line plot comparing the daily performance of the portfolio against the S&P 500.



The third graph plots the the mean versus standard deviation of each stock performance. This plot is useful in examining the trade off between the expected return and the risk of the portfolio. The program uses each stock's daily performance, i.e., the daily percentage change in price, to calculate the standard deviation and the mean.



The fourth plot is a heat map that displays the stock correlation.



The last one is a line plot that allows the user to compare the 50 days moving average price with the daily price of the portfolio.

