

# **STOCK CALL & PUT ADVISOR**

## **V1.0**

Program Summary & Version Notes

Phillip Korolev  
2023-10-29

## Program Summary

The principal function of the first version of the Stock Call/Put Advisor program is to showcase the day's top ten performing stocks, and top ten losing stocks. With a short analysis, and a volatility algorithm implemented in the program, it will then evaluate each of the stock's volatility and suggest three stocks from each category to either open a call or put position on, based on their evaluated score.

Program utilizes pandas, requests, io and yfinance libraries.

---

## Background Information

The price of all stocks varies throughout the day, however, some stocks see an extreme one-day surge; and others, a drastic one-day fall. This happens for a variety of factors, including the release of quarterly reports, company announcements, market updates, social trends, and more technical factors such as *On Balance Volume* or its *Relative Strength Index*. One trend we see in the majority of cases, is that these surges or falls are not long-lasting. For instance, Nikola Corp (NASDAQ: NKLA) had seen an extreme surge in price, jumping from 1.18USD to 1.71USD (44.92%), but since falling back to its natural price of roughly a dollar, 1.01USD (-40.94%). In this case, our program would have actually suggested the right action to take on the day of its surge: shorting the stock over the next couple of weeks. These extreme one-day movers are usually caused by short term high volume trades, occurring for no longer than a few days, which is why in most cases, the stock tends to fall/rise to its true value.

Oftentimes, checking the list of the top gainers or top losers stocks is enough to evaluate the right position to take, since we can infer that in most of the cases, an extreme one-day surge/fall in price is usually "faulty". However, there exists numerous factors that may dictate whether the price will continue to rise or fall. For this first version of the program, I have implemented a basic volatility equation, helping to evaluate the overall strength of each stock from the list of top ten gainers (one-day surging stocks) and top ten losers (one-day crashing stocks).

---

## Program Functions & Classes

### Gainer

This *class* enables us to create a **Gainer** object by calling `Gainer()`, where we can then create a data frame of the top ten gainer stocks of the day, and read this data with a variety of functions. Its simple constructor only defines `self`, and no other parameters.

- **`createDataFrame(self)`**: by using a url of a website containing top gainer stock information, we access it as a browser, wrapping the data in an IO String, then creating and returning our data frame with pandas, showcasing the day's top ten surging stocks.
- **`getTickers(self)` / `getPrices(self)` / `getDayChanges(self)` / `getMarketCaps(self)`**: returns a list of tickers or prices or daily price changes or market caps of each stock respectively.
- **`getDayChangesFloat(self)`**: returns a list of daily price changes as floats (00.00), instead of a string ('00.00%').
- **`getTickersAt(self, position)` / `getPricesAt(self, position)` / `getDayChangesAt(self, position)` / `getMarketCapsAt(self, position)`**: returns ticker symbol or price or daily price change or market cap at some position within its respected list; position ranges from 0 to 9.
- **`getDayChangeFloatAt(self, position)`**: returns daily price change at some position as a float; position ranges from 0 to 9.

### Loser

This *class* enables us to create a **Loser** object by calling `Loser()`, where we can then create a data frame of the top ten loser stocks of the day, and read this data with a variety of functions. Its simple constructor only defines self, and no other parameters. All functions mimic those in the Gainer class above.

### calculateVolatility(stock, dayChange)

This function takes in a stock ticker symbol as a string (e.g. 'AAPL') and the stock's one-day change percentage as a float (e.g. 34.72). Using the `yfinance` library, it fetches historical data of the imputed stock ticker symbol, and implements the volatility equation explained below. The function returns a volatility score as a float.

### main()

This function creates a **Gainer** and **Loser** object, along with their respective data frames and prints them. Then, it evaluates each of the stock's volatility, displaying their scores. Finally, the function suggests three stocks from each category to SHORT or BUY based on their volatility, and displays the execution time.

---

## Volatility Equation

For this project, I derived a simple risk or volatility equation that utilizes a stock's historical and current price and change % comparison. In this equation, we focus on a stock's six month change, and its recent one-day surge/fall. We calculate a stock's average daily change throughout the last six months by dividing its six-month price change (%) by roughly 126, the amount of trading days in a half-year. And, we will also define the stock's recent one-day change to create our two variables:  $c$ , our average one-day change in the last six months and  $d$ , our recent one-day change, to create our basic volatility equation:

$$vol = |(d - c)(\frac{d}{c})|$$

*\*In the program, the final volatility score is also divided by 100 to simplify the scale.*

From immediate observation, the equation scores high, if the average daily change is much smaller than the current one-day change, and low if the average daily change is similar to that of the current one-day change. This makes sense, since stocks that see a sudden drastic jump or fall in price are often labeled volatile or risky, because it does not match with its natural daily growth or decline. Again, we can use the equation to derive realistic arbitrary results.

We have four main cases for a stock's daily average move compared to its recent one-day move. Taking  $M > 0$ , an arbitrary large number to evaluate large one-day moves and  $m > 0$ , an arbitrary small number to help evaluate small one-day moves. Notice that in these arbitrary cases,  $d$  is a multiple of  $c$ , for instance,  $d$  can be a huge one-day increase in price, and  $c$ , a small average daily increase; we can evaluate  $d$  as  $Mc$ . In another case, we might see a huge one-day fall,  $-d$ , but a positive average one-day change,  $c$ , here, our one-day change value would be  $-Mc$ .

See all cases and proofs below...

	One-Day Change ( $d$ )	Avg. One-Day Change ( $c$ )	Volatility Score $ (d - c)(\frac{d}{c}) $
<i>HIGH VOL</i>	$Mc$	$c$	$ (Mc - c)(\frac{Mc}{c})  =  c(M - 1)(\frac{Mc}{c})  =  (M - 1)(Mc) $ Since $M$ is an arbitrary large number, we do not account for the -1 in $(M - 1)$ , so we have, $ (M - 1)(Mc)  =  (M)(Mc)  = M^2c$ Finally, we get $M^2c$ , a large value, indicating a very volatile stock, as expected.
	$-M(-c)$	$-c$	$ (-M(-c) - c)(\frac{-M(-c)}{c})  = \dots =  -M^2c  = M^2c$
	$M(-c)$	$-c$	$M^2c$
	$-M(c)$	$c$	$M^2c$
<i>LOW VOL</i>	$mc$	$c$	$ (mc - c)(\frac{mc}{c})  =  c(m - 1)(\frac{mc}{c})  =  (m - 1)(mc) $ Since $m$ is an arbitrary small number, we can account for the -1 in $(m - 1)$ , so we have, $ (m - 1)(mc)  = m^2c - mc$ , Finally, we get $m^2c - mc$ , which is a small value, indicating a very stable stock, as expected.
	$-m(-c)$	$-c$	$ (-m(-c) - (-c))(\frac{-m(-c)}{-c})  = \dots =  (m + 1)(mc) $ Since $m$ is an arbitrary small number, we can account for the +1 in $(m + 1)$ , so we have, $ (m + 1)(mc)  = m^2c + mc$ , We get $m^2c + mc$ , still a small value, indicating a low-volatility, but still greater than the previous case, since this time, the one-day change is in the opposite direction of the average one-day change, which adds volatility, so our equation checks out once again.
	$m(-c)$	$-c$	$m^2c - mc$
	$-m(c)$	$c$	$m^2c + mc$

## Example Output & Result Analysis (2023-10-29)

```
Top 10 Gainers Info:
No. Symbol Company Name % Change Stock Price Volume Market Cap
0 1 FRBN Forbion European Acquisition Corp. 59.33% 28.44 249585 449.71M
1 2 SPEC Spectaire Holdings Inc. 45.34% 2.34 4118197 46.15M
2 3 NRDS NerdWallet, Inc. 40.66% 9.34 7682681 724.20M
3 4 IMTE Integrated Media Technology Limited 29.32% 2.47 305195 5.27M
4 5 COCH Envoy Medical Inc. 23.89% 1.40 51797 29.82M
5 6 WRAP Wrap Technologies, Inc. 23.40% 1.74 342367 75.20M
6 7 IBRX ImmunityBio, Inc. 20.98% 2.48 33114047 1.66B
7 8 WBUY WeBuy Global Ltd 19.61% 4.94 311144 255.95M
8 9 DECK Deckers Outdoor Corporation 18.94% 576.37 1821341 15.06B
9 10 BTDR Bitdeer Technologies Group 18.50% 3.78 580264 420.45M

Top 10 Losers Info:
No. Symbol Company Name % Change Stock Price Volume Market Cap
0 1 LTRPB Liberty TripAdvisor Holdings, Inc. -57.97% 10.50 52822 50.07M
1 2 AIRE reAlpha Tech Corp. -49.63% 23.17 89631 985.20M
2 3 SYTA Siyata Mobile Inc. -47.73% 0.67 1690880 1.25M
3 4 BWMX Betterware de MÃxico, S.A.P.I. de C.V. -38.20% 10.55 346696 393.69M
4 5 BHAT Blue Hat Interactive Entertainment Technology -37.69% 1.62 2711943 92.02M
5 6 TRCA Twin Ridge Capital Acquisition Corp. -36.25% 7.65 85210 86.65M
6 7 JWEL Jowell Global Ltd. -35.03% 2.63 64313 89.88M
7 8 VVOS Vivos Therapeutics, Inc. -33.33% 4.32 194438 129.29M
8 9 B Barnes Group Inc. -32.82% 20.14 2169099 1.02B
9 10 LRHC La Rosa Holdings Corp. -28.40% 1.21 118962 8.47M

Evaluating Gainer Stock Volatility...
FRBN : 352.00488999999993
SPEC : 35.33702480786169
NRDS : 45.623585704450626
IMTE : 20.21431493706736
COCH : 8.476039770523773
WRAP : 63.99900235605247
IBRX : 12.032730373064565
WBUY : 8.751253189411763
DECK : 84.374711269777
BTDR : 6.95241370859845

Evaluating Loser Stock Volatility...
LTRPB : 315.38858294647036
AIRE : 2.2985642442056538
SYTA : 32.52744018189114
BWMX : 35.35380377644018
BHAT : 7.513004402246529
TRCA : 76.15849759360134
JWEL : 99.98479855391057
VVOS : 63.455420444533345
B : 56.60215571177878
LRHC : 17.413899462712575

Evaluating Appropriate Stocks to SHORT Based on Highest Volatility...
FRBN | Price: 28.44 | % Change 1D: 59.33% | Market Cap: 449.71M | Volatility: 352.00488999999993
DECK | Price: 576.37 | % Change 1D: 18.94% | Market Cap: 15.06B | Volatility: 84.374711269777
WRAP | Price: 1.74 | % Change 1D: 23.40% | Market Cap: 75.20M | Volatility: 63.99900235605247

Evaluating Appropriate Stocks to BUY Based on Highest Volatility...
LTRPB | Price: 10.5 | % Change 1D: -57.97% | Market Cap: 50.07M | Volatility: 315.38858294647036
JWEL | Price: 2.63 | % Change 1D: -35.03% | Market Cap: 89.88M | Volatility: 99.98479855391057
TRCA | Price: 7.65 | % Change 1D: -36.25% | Market Cap: 86.65M | Volatility: 76.15849759360134

Execution time: 18.64181399345398 seconds
```

From October 29<sup>th</sup>'s suggestions, we see that a short (put) on FRBN, for example, would have been successful, as the stock dropped the following day to 21.10USD (-25.81%). From the same suggestion, a buy (call) position on JWEL would have been successful, as the stock has increased in price since its one-day plummet.

### **Future Features**

In the following versions of the SCPA program, it's necessary to add more factors that may influence the program's suggestions as to which stocks to short or buy from the two categories. These factors include OBV (On Balance Volume) and Relative Strength Index, and could be implemented in different functions (e.g. `calculateOBV(stock)`, `calculateRSI(stock)`).

Stocks should also hold more information about themselves, such as open price, close price, day low, day high, volume, average ten-day volume, total debt, gross profits, etc. This information can be stored in variables, or calculated upon request by fetching data from the `yfinance` library, and would help create a more appropriate suggestion by the program.