

# STOCK-CHART

COMPARE STOCKS AND ETF URL <https://stocks-compare.netlify.app>

Stock-chart is used to compare stocks and ETF in American stock market NYSE and NASDAQ. The tool is designed for long term investors.

## USAGE AND TESTING INSTRUCTIONS

- Load URL to browser <https://stocks-compare.netlify.app>  
First load, A sample list of stocks will be loaded .
- Press `<fill_gain_info>` to fill gain and info, which other users shared through Firebase.  
The information may be old. Check the columns info\_date & gain\_date
- Press `<info>` button, to bring overview data from AlphVantage.co (Not for ETF)  
Checkbox `<StockInfo>` shows details
- Press `<gain>` button, to bring gain data from AlphVantage.co. Pressing `<gain>` draws a chart.  
Gain == 2 means, 100% price increase.
- Hide/Display columns by selecting or deselecting check boxes at the top
- Sort a column by pressing the column title. Second press will reverse order.
- Press `<del>` to delete a stock symbol
- Add a stock or ETF symbol, and press `<Add>`
- `<Show-stocks-compared-to-QQQ-5y>`
- Global Search/filter string. Show only rows that contain the text.
- Persistence: Press `<save-data>` button. Then close the browser tab, and reload again. Verify same data.
- The chart uses Plotly library. It allows zooming, and revert zoom.
- Manual: check the `<Manual>` checkbox at the bottom.
- `<Drop-Recovery>` see below

## CHART

`<chart>` checkbox allows to hide/show chart of a stock or ETF (hide by default)

Each press of `<gain>` will prepare a gain chart of the symbol. Up to 22 years based on weekly opening price.

Chart with calculated splits, will be purple, otherwise green. (No splits)

## ALPHAVANTAGE.CO RAW DATA (GAIN AND INFO) SOURCE

The tool gets the raw data from AlphVantage.co API. The user needs to get his own <key> from AlphVantage.co site. The key should be entered in the <config> pan. Otherwise you are sharing my key.

Free key allows 5 API calls per minute, and 500 per day. Please get your own <key> from AlphVantage.co site.

## **ALPHAVANTAGE.CO API GAIN PITFALL**

Before relying on the tool, the user MUST compare the values with other providers like, google finance, or finance.yahoo.com

Market info retrieved through AlphaVantage.co API. The API provides nominal stock historical prices and ignores splits and reverse-splits, and provide nominal stock or ETF price. The tool get the splits from <https://www.stocksplithistory.com/> and calculate compensation of splits. The button <calc-splits> allows to bypass these calculations. And get raw prices

## **SPLITS**

Splits\_retrieved from <https://www.stocksplithistory.com/>

can be reviewed at column <splits\_list> in the table

## **VERIFY\_1**

Values close to 1 means the gain values are good.

Variations from 1 happens because of date difference, and close/open market difference.

When pressing the <gain> button the tool retrieves the price from [bigcharts.marketwatch.com/historical/default.asp?symb=IBM](https://bigcharts.marketwatch.com/historical/default.asp?symb=IBM)

And compare it with alphaVantage value

Comparison info like date, price can be seen by pressing <verifyColumns>

## **DROP-RECOVERY**

The tool analyses stock drop during market drop. Shows drop-level, date of deep price, and stock recovery duration in weeks. The tool is valid only during an overall bull market, with some bear periods. The tool is limited to 22 years of AlphaVantage data. Default is last 8 months.

Steps:

- Open Drop-Recovery pan, by pressing <Drop-Recovery> check box.
- Choose a date range of the drop (Or big correction)
- The tool calculate 3 related columns:
  - Deep: how low did the stock drop from highest price before drop. (0.7 means a drop of 30%)
  - Recover week: how many weeks it took to reach the pre correction stock price.

- Drop bottom week date approximation.
- Press the **<gain>** button of some stocks.
- Algorithm
  - Search for the lowest price (deep) within selected dates
  - Search for highest price before correction
  - Search for recovery duration in weeks
  - Calculate deep, and recovery duration.

Default dates are: 2008 market drop, and Mar 2020 market drop.

Choose to correction periods **<2008>** **<2020>** **<last 8 months>** or any valid date.

## **SHOW-STOCKS-COMPARED-WITH-QQQ**

Press **<Show-stocks-compared-to-QQQ-10Y>** buttons show a list of stocks and ETF that had a yearly gain equivalent or better than QQQ, the list is retrieved from shared database (firebase), that other users examined **<gain>**

The bench mark for comparison is QQQ ETF, NASDAQ hi-tech average. User should hold only stocks that perform better than QQQ, or invest in QQQ. The columns: wk, mon, 3mon, yr, 2yr, 5yr, 10r, shows the gain proportion. A gain of 2, means the stock price doubled during that period. Other fields can be found on google.

## **<FILL-STOCKS-COMPARED-WITH-QQQ>**

Press **<Fill-stocks-compared-to-QQQ-5y>** button to insert in table stocks and ETF that are equivalent or better than QQQ

## **BACKEND DATABASE (FIREBASE). SHARE INFO BETWEEN USERS**

The tool stores gain and info data in a global database, to be shared with other users. This data is used to fill gain and info in the table.

## **PERSISTENCE**

The tool saves the table data in local-storage, for next session.

## **TABLE FIELDS**

**Symbol:** ticker of stock of a company or ETF

**Exchange:** NASDAQ or NYSE

**Industry:** Services / Electronic/ Medical / Biological / Computers/ Pharma

**PE**: Stock price / Yearly earning per stock

**PE(F)**: Forward PR

**PE(TTM)**: PE of Trailing 12 Months

**PEG**: PE / Yearly earnings growth

**Div**: Yearly dividend per stock / Stock \_price (0.05 means the that for \$100 worth of stocks, you get \$1)

**BETA**: fluctuation compared with other stocks of same industry (0.8 means less volatile than average)

**EV/EBITDA**: Enterprise value (Current Market Cap + Debt + Minority Interest + preferred shares – cash) / (earnings before interest, taxes, depreciation, and amortization)

**EV/Revenue**: Enterprise value / yearly revenue

**PriceToBook**: Company valuation / Book value.

**Target**: Analysts expected price / actual price

Wk - gain/drop during last week. (1.5 means rise 50%)

2wk - gain/drop during last 2 week.

Mon - gain/drop during last mon.

... 3mn, 6mn, Yr, 2yr, 5yr, 10yr, 20yr

**Splits**: number of splits, (Calculated when **<gain>** pressed)

**Split list**: split list of splits or reversed splits.

(retrieved from [www.stocksplithistory.com/&symbol=MSFT](http://www.stocksplithistory.com/&symbol=MSFT))

**Info date**: Date when info retrieved **<info>** button pressed)

**Gain date**: Date when gain retrieved **<gain>** button pressed)

**Deep**: Lowest price during drop / highest price before drop (Calculated when **<gain>** pressed)

**RecoveryWeek**: number of weeks to reach price before drop

**Deep date**: estimated week of lowest price.

priceDivHigh - cruurentPrice / highestPrice (Calculated when **<gain>** pressed)

verify\_1 compare oldest price with [bigcharts.marketwatch.com/historical/default.asp?symb=IBM](http://bigcharts.marketwatch.com/historical/default.asp?symb=IBM)

(Calculated when **<gain>** pressed)

verify fields: AlphaDate, alphaPrice, verify\_1

## **PLATFORMS**

The tool was tested on some browsers (Chromium, Firefox, Opera, and Microsoft Edge)

Preferred platform is desktop, laptop, or tablet. Not recommended for smartphone

## **USED PACKAGES**

The tool is built on top of JavaScript, React.JS library, react-table, Plotly, react-datepicker, Firebase, Redux,

bigcharts.marketwatch.com, <https://www.stocksplithistory.com/>

## **BUG REPORT AND ENHANCEMENTS**

The tool was developed by Eli Shagam, Please send any bug report, or enhancement request to [eli.shagam@gmail.com](mailto:eli.shagam@gmail.com)

## **Verify 1 Exceptions / Problems**

NVAX verify\_1 = 20 bigcharts.marketwatch.com miss NEGATIVE split Of 0.05

ACLS verify\_1= 3.17 bigcharts.marketwatch.com miss NEGATIVE split 0.33333

COPX verify=3.17 bigcharts.marketwatch.com miss NEGATIVE split 0.3333

VOO verify=1.98 bigcharts.marketwatch.com miss NEGATIVE split 0.5

CNC verify=1.25 bigcharts.marketwatch.com correction 12?, splits 2\*2\*2