Title: Stock Analysis and Visualization
Authors: Henry Fox, Syed Hani Haider, Shagun Jindal
Github: https://github.com/hanihaider1/Stocks

Summary:

The purpose of this package is to provide the user with basic equity performance statistics and a graphical representation of a user selected publicly traded company's share price over time. Once the user downloads the selected company historical share price data from Yahoo finance which should be saved in the same directory as the package is saved in, our program will conduct a simple table of the company's performance during the time period.

For this project we used three external libraries. The data analysis libraries Pandas and Numpy will be used for creating the summary table for the data and importing the csv files. The data visualization library Matplotlib will be used for visualizing the stocks data graphically. Over the past 10 years python has made its mark in the financial markets and has streamlined how data is shaping the investment decisions of many savvy investors. There are many modules and packages that are like what we have developed for this project. We believe in this crowded space there is a need for a simple package that supplies the facts and provides insight for the financial decision maker which is what we believe our package delivers.

Design:

The design of this package will center on creating a strong framework for any programmer that wishes to utilize or customize this package to aid in making their investment decisions. We aim to deliver a readable and programmer friendly package that can produce the desired output as well as having a robust error handling safety net to notify the user of data or user input error. The following is an outline of our library:

1. *Summary Table*: The summary table will include a brief outline of the company's stock performance: when the company's stock price hit its highest high and its lowest low for the period, and the largest percentage change up and down during the time frame with the associated dates.
2. *Graphs*: The graph will be a line graph to help visualize the company's stock performance for the chosen period.
3. *Moving average*: The moving average is a layered feature on the graph which will be limited to the 50- and 200-day moving average for the company's stock. For many investors these moving average provide technical indicators of the stock rising or falling.
4. *Comparing stocks summary with accompanying graph*: If the user decides to input two different publicly traded company stocks the table and graph will reflect the above information for both companies which will give the user a side-by-side comparison of the two companies.

Usage:

This package can be used by the novice or more advanced financial investor. The *Stock Analysis and Visualization* package would be the first step for the investor who is conducting an initial search of a large range of publicly traded companies that fit his or her investment goal. Once the code runs the user is presented with a graph (*image 1*) that provides a graphical representation of the company's performance with the accompanying 50 and 200-day moving averages.

Also provided is a summary table (*image 2*) that is an overview of the investment performance during the time period. The combination of these two outputs will provide the investor a tangle idea of the highest high, lowest low, and overall equity performance of the company. To provide the user with as much information as possible we have provided a comparison table (*image 3*) that contains the target companies stock performance data as well as its closest competitors. With these outputs we believe that the investor would be informed enough to make an investment decision to buy or sell the company's stock.

Image 1

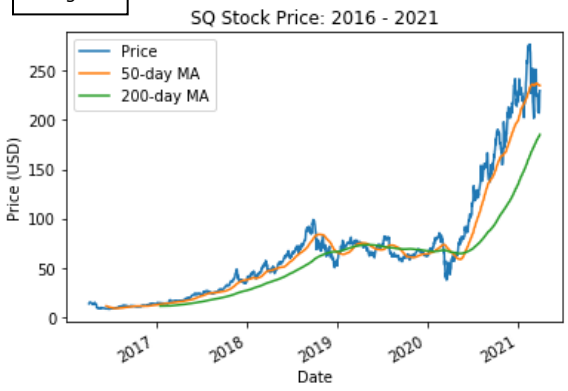

Image 2

summary	Highest High	Lowest Low	Largest Move Up in a single day %	Largest Move Downward in a single day%
Price	276.57	38.09	9.169	-10.4509
Date	2021-02-19	2020-03-20	2020-04-08	2020-07-13

Image 3

summary	Highest Price	Lowest Price	Percentage Change %
SQ	276.570007	8.620000	10.185825
PYPL	304.790009	34.200001	8.742628

Discussion:

Our package is most like the library QuantPy which provides a framework for quantitative finance with python. Some current capabilities are the importing of daily returns from Yahoo Finance, the calculation of optimal weights for Sharpe ratio and efficient frontier, and event profiler (*Kumar, S.*). Our package would not directly compete with QuantPy users because our package provides a simpler output and does not provide as many trading signals as QuantPy.

Users would be drawn to our package if they wanted a quick and legible macro pictures of the company's performance while ignoring complicated trading data or signals that might confuse the user. Our trading package can be improved by streaming live trading data into the package so the user would not have to download individual csv files and reducing the investment decision timeline.

Statement of contributions:

The group split the package into even sections so that each member of the team contributed equally to the successful delivery of the assignment. Henry Fox wrote the code for the graphs and moving averages as well as the configuration of the graph to ensure readability and clean graphical output. Henry also wrote and edited the proposal and final report.

Shagun worked on the comparison of stocks for SQ and PYPL based on the adj close data. A clean representation of the same can be seen through a graphical comparison between the two stocks. She also worked on comparing stocks summary which returns the highest price, lowest price and percentage change for the two stocks and is represented in form of a table.

Hani was tasked with creating the error handling module and the summary tab. To capture any common errors a user might make Hani setup an error tracking module that would catch common errors such as missing data, the user uploading a non-csv file and missing headers. For Hani' second task he used the Numpy library to create the summary table.

References:

1. Kumar, S. (2020, June 07). Best Python Libraries/Packages for finance and financial data scientists. Retrieved April 19, 2021, from https://financetrain.com/best-python-librariespackages-finance-financial-data-scientists/