Stat 461 Final Project Report

Xiangyu Zeng

Introduction & Objective

A portfolio is a combination of a multitude of financial assets such as stocks and options. It typically consists of an unequal distribution of investor's money into different assets with the goal of making a positive return and creating wealth. With a perfect mix of assets or investments, a well-diversified portfolio can meet the required return on investment expected by the investor with the minimum risk possible. With final goal of making successful portfolio investments on the financial stock market, in this project I propose to make thorough financial analysis on selected stocks from four renowned tech company stocks, Apple(AAPL), Google(GOOG), Microsoft(MSFT), Oracle (ORCL) by fully applying the knowledge I learnt from the lectures. Preliminary financial analysis with regarded with data transformation, model fitting, forecasting, call & put option investment and portfolio returns is made before I compute the final model for portfolio investments. Each section is bulleted and written in the corresponding orders.

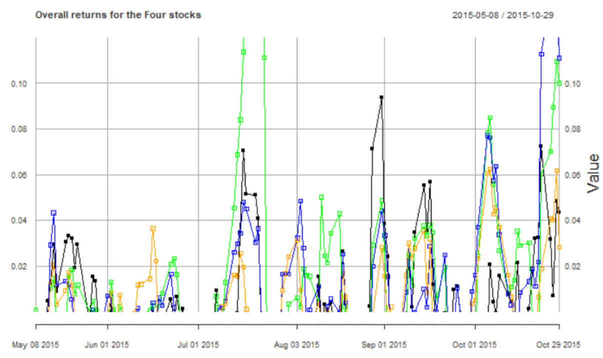
Data description

The historical data for individual company are directly obtained from R using "GetSymbol" command. For computing purposes, the stock comparison's csv file that consists of combined 10 year historical stock price for those four companies is, on the other hand, downloaded from Yahoo Finance, a financial database that including statistics for various financial assets. In particular, my data set consists of a total of 10,068 daily adjusted stock price from each of those four companies ranging from 2005-10-31 to 2015-10-31.

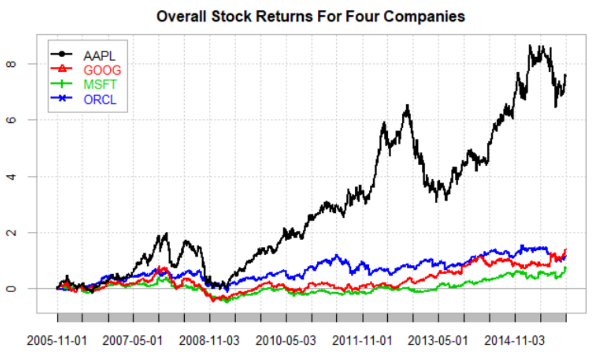
Observational and Exploratory analysis on Four stocks

To begin with, I plotted out the returns based on adjusted prices across four companies. It can be seen from the graph that among the four companies, google has the widest fluctuations, highest stock price per share and returns with respected to other three companies. Base on this result, I decide to conduct in-depths analysis with google stock data set, which will be recounted with details in later sections. [Please note that the same reasoning process also applies to my analysis on other three companies, due to constraints of paragraphs I omit them].

Picture 1

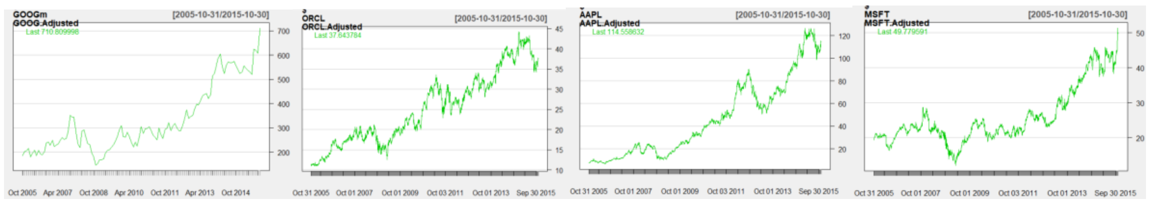


Picture 2



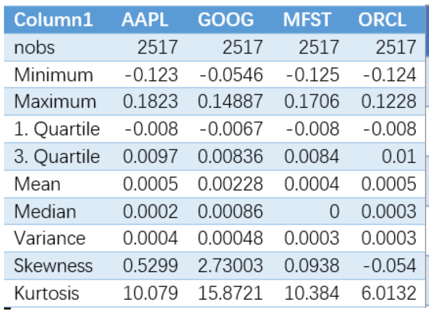
From the overall returns and line index returns(picture 1&2), I find that Oracle and Apple’s adjusted prices seem to have steadily increased over time. Conversely, Microsoft and Google’s stock price seem to significant decrease during the year 2007-2009, and then begun to rise again. One of the possible reason might be due to the influences from the 2008 global financial crisis. In general, google stock has trends of steep increase, an indication of its strong investment potentials.

Picture 3-6



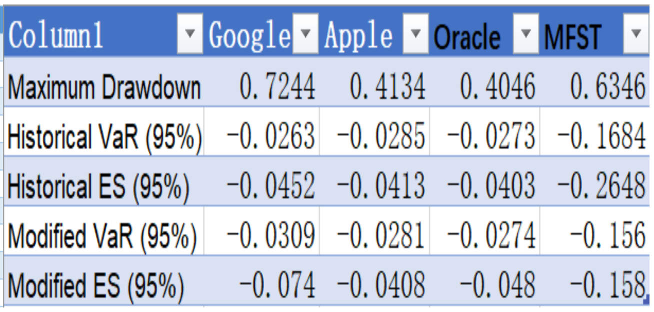
From each company’s individual stock plot (picture 3-6), I observe serious heteroscedasticity problem as shown by the random spikes in their individual plots. This suggests for a GARCH model fit for the data set. As discussed in later sections, all original price data are not stationary and need for log transformation. Moreover, important properties like symmetry, tail behavior, normality and outlier all need to be tested. Therefore, I compute the basic statistics for the transformed log returns and conduct hypothesis test with regarded with all properties mentioned above. The details are in the R code attached and the output table(picture 7) is displayed below. This summary table of model properties is essential to the GARCH modeling fitting section that will be discussed in later sections.

Picture 7



Column1	AAPL	GOOG	MFST	ORCL
nobs	2517	2517	2517	2517
Minimum	-0.123	-0.0546	-0.125	-0.124
Maximum	0.1823	0.14887	0.1706	0.1228
1. Quartile	-0.008	-0.0067	-0.008	-0.008
3. Quartile	0.0097	0.00836	0.0084	0.01
Mean	0.0005	0.00228	0.0004	0.0005
Median	0.0002	0.00086	0	0.0003
Variance	0.0004	0.00048	0.0003	0.0003
Skewness	0.5299	2.73003	0.0938	-0.054
Kurtosis	10.079	15.8721	10.384	6.0132

Picture 8



Column1	Google	Apple	Oracle	MFST
Maximum Drawdown	0. 7244	0. 4134	0. 4046	0. 6346
Historical VaR (95%)	-0. 0263	-0. 0285	-0. 0273	-0. 1684
Historical ES (95%)	-0. 0452	-0. 0413	-0. 0403	-0. 2648
Modified VaR (95%)	-0. 0309	-0. 0281	-0. 0274	-0. 156
Modified ES (95%)	-0. 074	-0. 0408	-0. 048	-0. 158

Next, I dip into a more technical aspect on analyzing stock performances. I conduct backtest in R using the help from R's TTR package. This package allows me to create a DVI indicator (created by David Varadi) for each stock. Next, I use a threshold of 0.05 for the signal as stock indicators. Base on this result, I then calculate signal-based returns (today's percentage return multiplied by yesterday's signal and divided by position size). Finally, the PerformanceAnalytics package in R allows me to create a summary table(picture 8) with regarded to drawdowns, Value at risk(VaR), downside risks, and all other relevant statistics that assess the stock performances. Base on this table, I find Microsoft's stock performs relatively well and is a comparatively safer stock investment option. The parameters of other three stocks all share close values, which stop me from making further interpretations.

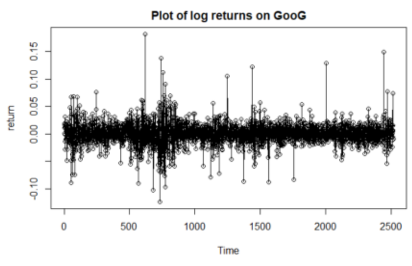
Google Stock Performance Analysis

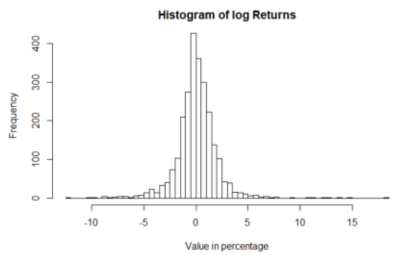
As mentioned before, I conducted in-depth analysis with google stock data set, the same reasoning process applies to other 3 stocks and is omitted due to word constraints.

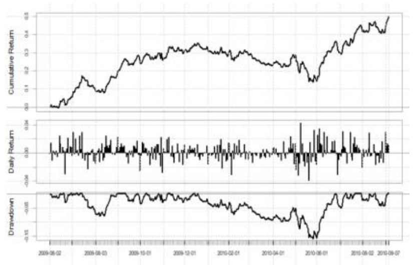
1.Data Transformation & Reasoning

From google's time series plot on adjusted price, I notice that prices are not stationary whereas the plot for the returns does appear to be stationary (i.e. the data does fluctuate around a common mean or location). The p-value from the kpss.test() statistics also confirm my findings. Therefore, I decide to perform a log transformation (i.e taking the natural log of the quotient of consecutive observations) on the non-stationary data set. Next, I use basicStats() function to obtain summary statistics on the log returns. from the return plot I see that there exists heteroskedasticity in the model. However, I cannot determine whether this is enough to warrant consideration. I will begin with a simple model and work my way to more complex models when and if the simple models are poor fits to data. Note: The same general reasoning applies when I analyze the other 3 stocks. As a result, I perform log transformations on all three stocks.

Picture 9-11



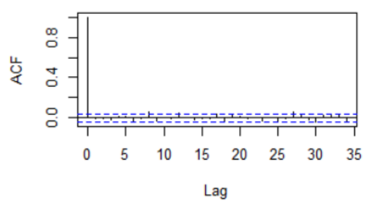


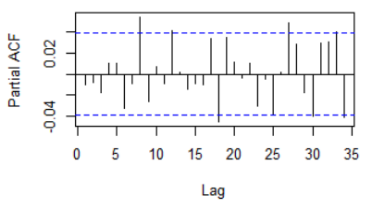


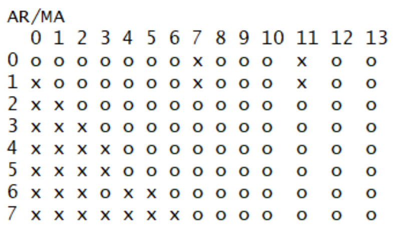
2. Model Fitting

Next I construct the ACF and PACF plots. The ACF plot has few spikes within the confidence range while the serial correlations do not appear to be significant. The PACF shows similar results but with some slightly significant correlations. In general, there are no obvious patterns from my observations. Meanwhile, to test for ARCH effect, I also plot out ACF and PACF for squared returns. The results have many spikes out of the confidence range, an indication of significant correlations between one and another. Garch model is needed to remove such effects.

Picture 12 - 14



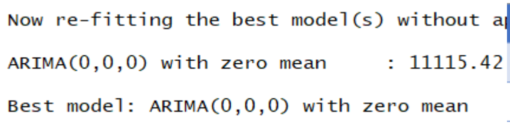




```
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 o o o o o o o x o o o  x  o  o
1 x o o o o o o x o o o  x  o  o
2 x x o o o o o o o o o  o  o  o
3 x x x o o o o o o o o  o  o  o
4 x x x x o o o o o o o  o  o  o
5 x x x x o o o o o o o  o  o  o
6 x x x o x x o o o o o  o  o  o
7 x x x x x x x o o o o  o  o  o
```

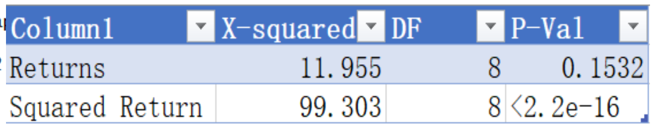
To help determine the model, I then take some steps to make in-sample comparison. I construct a matrix (with orders of 6x6) to store pairs of Akaike information criterion (AIC) and Bayesian information criterion (BIC) scores and the lowest ones will suggest order for my model. AIC suggests model ARMA(2,3) while BIC suggests ARMA(0,0). Base on the results of the residual plot, I decide to go with ARMA(0,0) because the plot of the residuals are, in contrast, closer to 0, variance more time invariant and serial correlations not as significant. Further evidence from EACF table(picture 14) also confirms my findings as the starting point of the triangle shape is (0,0). After researching online, I found that auto.arima() function that also returns the same result(picture 15). This helps confirm my conclusion on model determinations.

Picture 15



```
Now re-fitting the best model(s) without a
ARIMA(0,0,0) with zero mean      : 11115.42
Best model: ARIMA(0,0,0) with zero mean
```

Picture 16



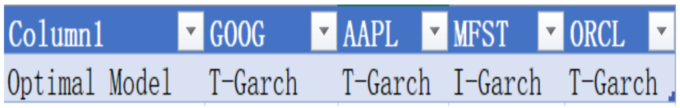
Column1	X-squared	DF	P-Val
Returns	11.955	8	0.1532
Squared Return	99.303	8	<2.2e-16

As mentioned before, from the plot I observe significant ARCH effect on squared residuals. To confirm my findings, I apply Ljung Box test (with null hypothesis of $\rho 0 = \rho 1 = ... \rho 8 = 0$) and find extremely small p value, an indication strong serial correlations(picture 16). Therefore, I further consider fitting GARCH model to remove such ARCH effects.

At first try I attempt to fit standard normal Garch model. There are still minor significant serial correlations in the standardized residual time, but those effects can be removed by extending the ARMA part of the model. At this stage, I think the evidence against simple ARMA(0,0) model is not that strong with those minor violations. The GARCH(1,1) model is adequate in explaining serial correlation in variance part as there is no significant serial correlation in the squared standardized residual time series.

Further, I find that the distributional assumption can be improved by considering some heavy-tailed distributions (student t distribution). Hence, I attempt to fit ARMA-IARCH(1,1),ARMA-TARCH(1,1) and ARMA-GARCH-M(1,1) model, the details can be seen from the R code. In the previous section, I had computed a summary property table for each stocks, now I could test the hypotheses of symmetry, normality and normal kurtosis to examine the basic assumptions with regarded to garch model. The results are shown in table below. Basically all stocks rejects these hypotheses(except tail behavior on apple stock), so I conclude that basically all log returns are asymmetric and have heavy tails. Thus I needed to consider skewed student t distribution when fitting my model. Under those assumptions, I attempted to fit a model to each stock individually. By comparing likelihoods for each of these models, I find that every model has a large p-value for Pearson goodness of fit tests. However, each model other than the tGARCH model rejects the null hypothesis in the sign-bias test, suggesting a poor fit to the data. However, this is not true for Microsoft stock data. I-Garch model fits better in this case. Therefore, T-Garch model provides the best model fit for most of the stocks.

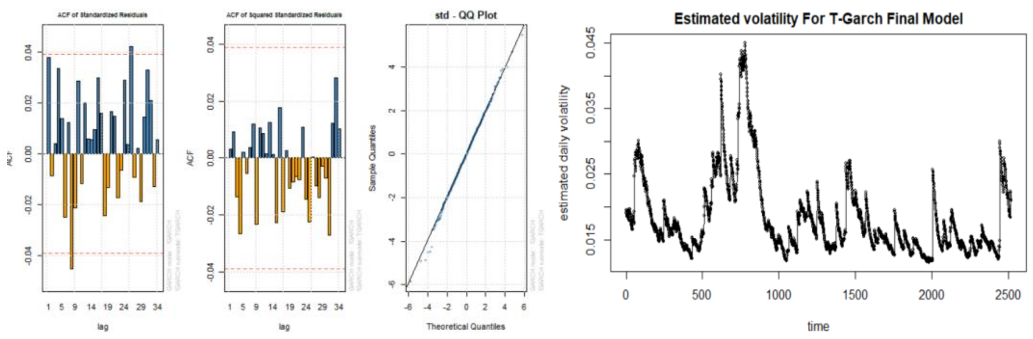
Picture 17



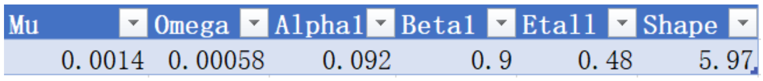
Column1	GOOG	AAPL	MFST	ORCL
Optimal Model	T-Garch	T-Garch	I-Garch	T-Garch

Now for google stock, I am interested in checking the distribution of the model’s residuals and Q-Q plot of the residuals that is shown below. Due to the fact that they follow a straight line pattern, I concluded that this model fits the returns pretty well. The ACF and PACF do not suggest significant correlations as well.

Picture 18-21



The final model parameters for this model and final equation is given below:



Mu	Omega	Alpha1	Beta1	Eta11	Shape
0.0014	0.00058	0.092	0.9	0.48	5.97

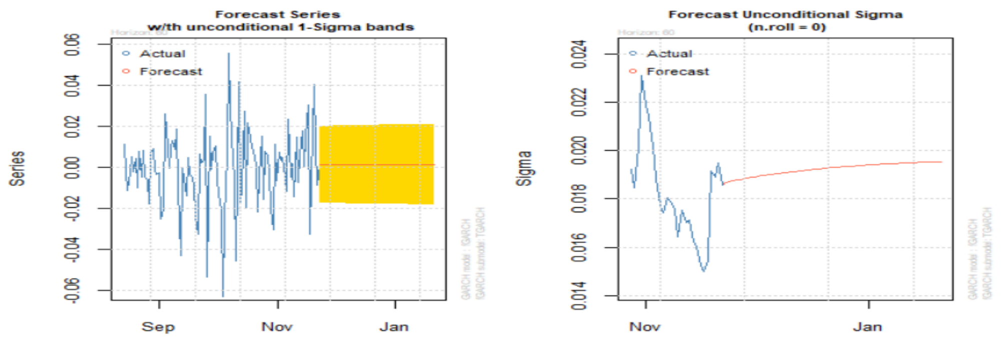
$$\sigma t^2 = 0.00058 + \sum_{i=1}^{s}(0.00058 + 0.48 * N_{t-i})a_{t-i}^{\ 2} + \sum_{j=1}^{m} 0.9\sigma_{t-i}^{\ 2}$$ where N_{t-i} is an indicator variable such that $N_{t-i} = \begin{cases} 1 & \text{if } a_{t-i} < 0; \\ 0 & \text{otherwise.} \end{cases}$

3. Model Forecasting

1.Ugarchforecast Model Forecast

Using the ugarchforecast function discussed in class, I make 60 steps ahead forecast. With a model fitted to the google stock data, I can make predictions about future price trend. As shown by the fitted volatility plot of the T- GARCH model(picture 16), I see a potential large spike in the next 60 days. This matches the behavior of the 2015 year stock data later. The estimated volatility diagram (picture 17) also predicts a period of high variance, which does not quite match the historical data.

Picture 22-23

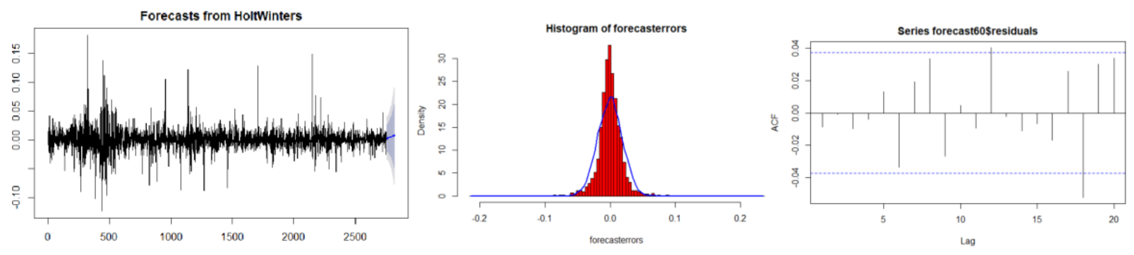


2.HoltWinters Model Forecast

The second method I use to make a predictive forecast with the google stock data set is the HoltWinters function in the “forecast” package. As the reports from previous section suggest, the google stock price can be represented as a general time series that can be described using an additive model with increasing or decreasing trend and no seasonality. After researching online for several forecasting models, I found that the Holt’s exponential smoothing functions most productive under this scenario when making short-term forecasts. Holt’s exponential smoothing estimates the level and slope at the current time point.

Smoothing is controlled by two parameters, alpha, for the estimate of the level at the current time point, and beta for the estimate of the slope b of the trend component at the current time point. With this method, I am not only able to make predictive analysis of the future trend(picture 24), but to acquire all relevant statistics regarded with forecast errors. Using the plotForecastError function, I am even able to graph a histogram of forecast error s(picture 25) where I can visualize the accuracy of my prediction.

Picture 24 - 27



The plot shows that the distribution of forecast errors is roughly centered on zero, and is more or less normally distributed, the acf of the forecast residuals is also plotted and there is no significant serial correlations between the forecast errors as can be confirmed by Ljung-Box test. This suggests that this HoltWinters exponential smoothing method provides an adequate predictive model.

Google Call and Put option analysis

1.The Black-Scholes model

This Black-Scholes model asserts that under no arbitrage assumptions, the price of European call option(c) and put option(p) has a closed form:

$$C = S_0\, e^{-qt} * N(d_1) - X\, e^{-rt} * N(d_2) \qquad d_1 = \frac{ln(\frac{S_0}{X}) + t\,(r - q + \frac{\sigma^2}{2})}{\sigma\sqrt{t}}$$

$$P = X\, e^{-rt} * N(-d_2) - S_0\, e^{-qt} * N(-d_1) \qquad d_2 = d_1 - \sigma\sqrt{t}$$

X is the strike price, t is the time to maturity, and N denotes the cumulative distribution function of the standard normal distribution.

From the Nasdaq Stock Market website, I can see various call and put options regarded with the stocks of the four tech companies that I analyze. Although I

already learn how to compute the value of options and how to arbitrage by hand from class, I am still interested in exploring the R functions that can help me calculate the results and make investment strategies on options.
I randomly choose a put option on a Google stock in June 2017 with a maturity of September 2017 (with 3 months of time to maturity). The current price of the underlying stock is USD 900, the strike price is USD 950, the volatility of Google is 22%, and the risk-free rate is 2%. I intend to calculate the value of the call option with the GBSOption() function from the fOptions package. Beyond the parameters discussed in class, I also have to set the cost of carry (b). After searching online, I find that in the original Black-Scholes model (with underlying paying no dividends), this equals the risk-free rate. The value of the put option is:

```
> GBSOption(TypeFlag = "p", S = 900, X =950, Time = 1/4, r = 0.02, sigma
= 0.22, b = 0.02)@price
[1] 67.05461
```

2.Cox-Ross-Rubinstein (CRR) model

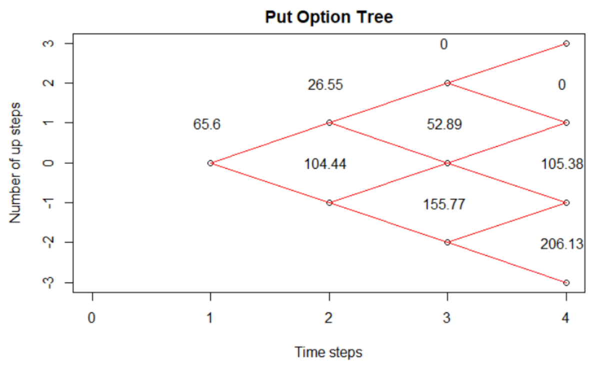
The Cox-Ross-Rubinstein model (Cox, Ross and Rubinstein, 1979) assumes that the price of the underlying asset follows a discrete binomial process. The price will either go up or down in each period and hence changes according to a binomial tree pattern, a concept that I had learnt in class.
To build the model, first I determine how many steps (n) I should model. Then I want to calculate u and d that measure the price change when it goes up and down. The final formula for calculating the call or put option value is :

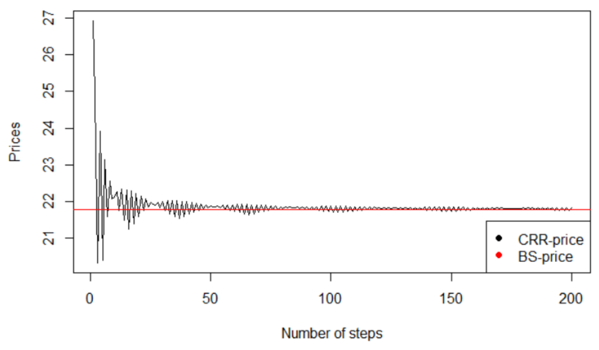
$$g = \left[p_n g_u + \left(1 - p_n\right) g_d \right] e^{-r\Delta t}$$

where g is the price of an option in a given node, gu and gd are the values of this derivative in the two possible nodes one period later.
Using R, I am able to compute the not only the value of the put option, but also the plot for the binomial tree process(picture 28 -29).

```
> CRRBinomialTreeOption(TypeFlag = "pe", S = 900, X = 950,
+   Time = 1/4, r = 0.02, b = 0.02, sigma = 0.22, n = 3)@price
[1] 65.59803
```

Picture 28-29





3.Model Comparison and Conclusion

From the R output I observed that the numerical results for the two models are close but not equal. I think the reason is that the stochastic processes that describe the price movements of the underlying asset are not identical. However, with increasing numbers of time steps the binomial process approximates the geometric Brownian motion. Therefore, the option price of the binomial model Should converges to that of the Black-Scholes model. Using R, I plotted out the prices determined by those two models with regarded with number of steps. The resulting graph attached above confirms my hypothesis.

The put option price on Nasdaq is 66.45, which makes it hard to analyze because different models give me different results on the predicted price of this option.

Final Portfolio investment analysis

In the last part of portfolio investment analysis, I intend to analyze two ways of computing the weights of each of the 4 stocks for the optimal portfolio. Both under the mean-variance model, one applies mathematical to solve for a final weight value while another offers an visual interpretation. Upon choosing the level of risk (volatility) the investors are willing to take, they can plug in the efficient frontier function to get the correspond weights between stocks.

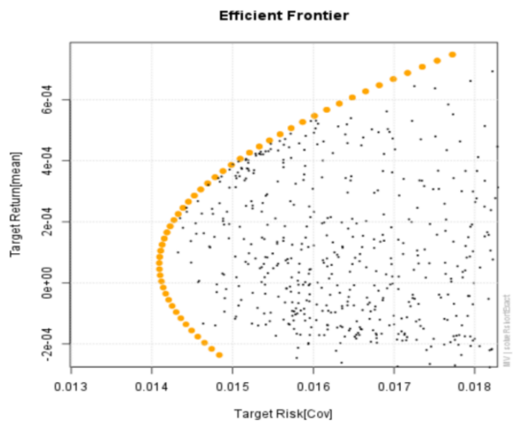
1. Mean-Variance model with efficient frontier visualization

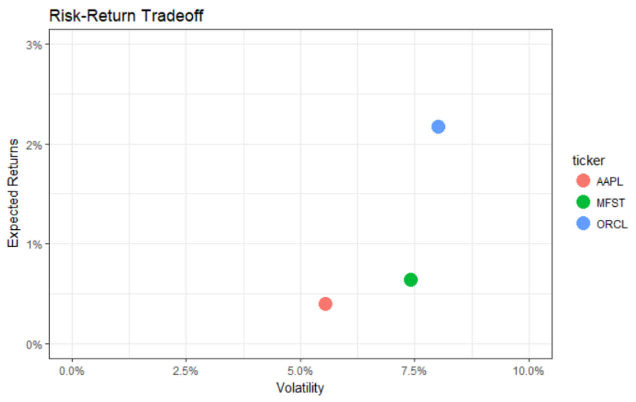
This method regards with plotting out the efficient frontier (desired return-minimum variance curve on the variance-return plane) to obtain the optimal portfolio. To be more specific, the efficient frontier represents the different combinations of weights of a given portfolio of assets. These set of portfolios either have the highest return for a predefined risk or on the opposite the lowest risk for a predefined return. Because my data set is too large with over 10,000 observations that exceeds R's graphing limits, I choose to simplify the stock comparison to 3 stocks instead of 4. The formula for expected return and expected standard deviation for the portfolio is as following: $\hat{r}_p = \omega_x \hat{r}_x + \omega_y \hat{r}_y + \omega_z \hat{r}_z$ where $\hat{r}_p$ stands for expected return, σ_p stands for expected volatility and ω stands for the weights.

$$\sigma_p = \sqrt{\omega_x^2 \sigma_x^2 + \omega_y^2 \sigma_y^2 + \omega_z^2 \sigma_z^2 + 2\omega_x \omega_y \sigma_{x,y} + 2\omega_x \omega_z \sigma_{x,z} + 2\omega_y \omega_z \sigma_{y,z}}.$$

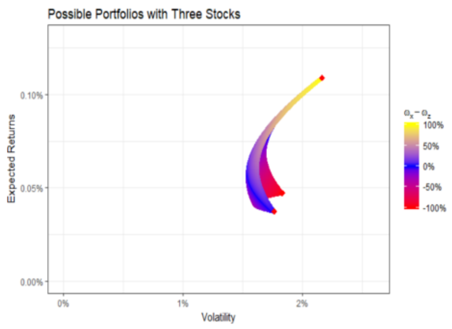
Using ggplot in R, I was able to compute the efficient frontier with respected to target risk(picture 30) and when I combine the new three stocks into a new portfolio, my portfolio has an expected risk-return tradeoff(picture 31). The crux is to calculate the values for a given 3 stocks and find an optimum portfolio

Picture 30- 31





Finally, using Scales and ggplot package I am able to plot out the efficient frontier diagram that consists of all possible portfolios, which has been expanded into a three dimensional diagram displayed below



Picture23

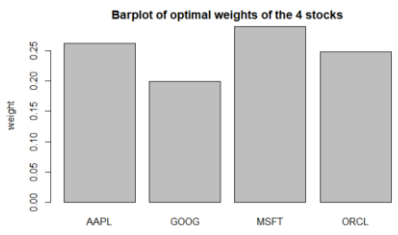
With the color coding, the upper point stands for stock x at ωx = ωz + 100%and the lower point stands for stock y (ωx = ωz - 100%). X,y and z are AAPL, MFST ,ORCL. All possible portfolios lay on this curve.

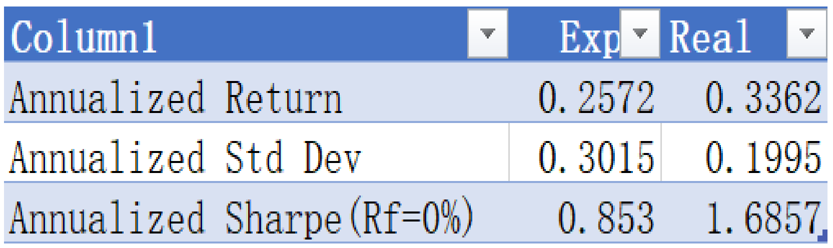
The colors in the picture above show the difference between the two different weights. A yellow color indicates a portfolio consisting mainly of the stock AAPL, a blue color indicates asset MFST, and a red area indicates a portfolio of asset ORCL. The three single asset-portfolios (the three red dots) is where the convergence happens. Next, using calcEFParams function in R I was able to get the values of $\alpha, \beta, \gamma, \delta$. Plugging in the parameters to the efficient frontier function $\hat{r}_{ef}(\sigma) = \frac{\beta}{\alpha} + \sqrt{\left(\frac{\beta}{\alpha}\right)^2 - \frac{\gamma-\delta*\sigma^2}{\alpha}}$, I can then get the optimal weights base on the value of risk I am choosing. Changing in accorded with different levels of risks the user am willing to take, this diagram of efficient frontier portfolio offers a more flexible way to visually interpret the Portfolio investment analysis.

2.Mean-Variance model with computation

A mean-variance portfolio is a portfolio with weights that has the lowest portfolio variance given a target return. In this method, the ideal weights between four stocks can be calculated with help from the portfolio.optim() function in the tseries package. The underlying assumption is that the target return equals the return on the equal weighted portfolio. Using the csv file that stores the combined stock prices of the four companies, I first created a vector of row mean returns and casted them into a xts object. Casting the results into the function, I was then able to compute the weights of each asset within the portfolio, the expected portfolio return and the standard deviation of the portfolio returns. A barplot (picture 33)with visual comparison between the weights of those four stocks and the relevant statistics are displayed below.

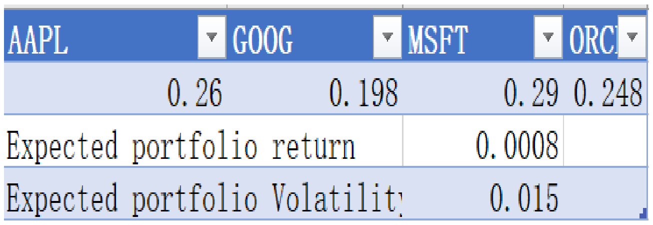
Picture 33-34





Column1	Exp	Real
Annualized Return	0.2572	0.3362
Annualized Std Dev	0.3015	0.1995
Annualized Sharpe(Rf=0%)	0.853	1.6857

As the output suggests, the portfolio computed with this method has an average return of 0.08% with a standard deviation of 1.5%. However, I can further optimize the target return by increasing the portfolio volatility(depends how much risk investors are willing to take). Under the hypothetical scenario that the investors want 10% greater target return compared with the average return, by computation I get the result that this leads to 8% increase in my portfolio's volatility. Finally, I compare the return I expected using the evaluation sample with the actual return on the out-of-sample period. From the resulting table, I can see there is a huge difference on the value of Sharpe ratio (average return earned in excess of the risk-free rate per unit of volatility). Since optimal portfolios contains the least volatility and the greatest Sharpe ratio, I learnt that this way might not be the most efficient way of computing. The portfolio final weight distribution diagram



AAPL	GOOG	MSFT	ORC
0.26	0.198	0.29	0.248
Expected portfolio return		0.0008	
Expected portfolio Volatilit		0.015	

Conclusion

In this project, I attempted to make financial analysis of four renowned tech company Apple(AAPL), Google(GOOG), Microsoft(MSFT), Oracle (ORCL) with respected their overall and individual stock performances over 2005-10-31 to 2015-10-31 time period. In particular, I closely analyzed google stock price performances while the same reasoning applies to the rest of three stocks. From the process of data transformation, model fitting, and forecasting, I make use of various R commands, packages and learn to offer visual interpretation on graphs of different financial models. I then consider the call & put option investment, base on the Black-Scholes model and Cox-Ross-Rubinstein (CRR) model, I analyzed an example google stock put option and come up with corresponding investment strategies. Finally, I analyzed two ways of computing the weights of individual stocks for the optimal portfolio. Both under the mean-variance model, one applies mathematical to solve for a final weight value while another offers a more flexible, visual interpretation. With the second method, I am able to compute the optimal portfolio weights, returns and relevant statistics on the selected 4 stocks for the final portfolio investment.

Appendices

Data Sources

http://www.nasdaq.com/symbol/goog/option-chain
https://finance.yahoo.com/quote/GOOG/
https://finance.yahoo.com/quote/AAPL/
https://finance.yahoo.com/quote/MFST/
https://finance.yahoo.com/quote/ORCL/
Quantmod package in R (use getSymbol command to get various data source)

Reference

https://www.r-bloggers.com/investment-portfolio-analysis-with-r-language/
http://blog.fosstrading.com/2011/03/how-to-backtest-strategy-in-r.html
http://economistatlarge.com/portfolio-theory/r-optimized-portfolio
http://www.quintuitive.com/2013/11/30/analyzing-the-dvi-indicator/
https://www.rdocumentation.org/packages/forecast/versions/8.1/topics/plot.forecas
t
http://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html
http://past.rinfinance.com/agenda/2009/yollin_slides.pdf

Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31:307-327, 1986.
Brockwell, Peter J. and Richard A. Davis. Time Series: Theory and Methods. New York: Springer–Verlag, 1987.
Michael, P. Introduction to R for Quantitative Finance. 2003.
Chatfield, C. The Analysis of Time Series, An Introduction", 2005.
Cryer T.Time Series Analysis with Applications in R, 2008

R code

The R code for my project is knitted to a separate pdf file and is turned in together with this project report.