Optimal Trading Strategy for Bitcoin and Gold

February 22, 2022

Stocks are renowned for constantly changing in price, spiking up and down and causing havoc for all traders. Whilst you can never be sure of your return, there are methods in which to take to help predict them. Starting with $1000, we aim to maximise our portfolio's value investing in both gold and bitcoin through 11/9/2016 - 10/9/2021. Accounting for the transaction fee for sales and purchases of bitcoin (2%) and gold (1%), we decide when to invest, sell and hold on to our assets in order to maximise our portfolio.

We started with a geometric brownian motion (GBM) model to calculate the drift parameter on each trading day and the price of the asset on future days. The drift parameter for certain days in predicting the path of bitcoin was significantly high resulting in over estimations of the price of bitcoin for that period. This wasn't as prominent in the gold path due to the low price of gold in comparison to the bitcoin. Despite the significantly high over estimations, the overall trend of the prediction fit the trend of the actual prices well.

Using the GBM model, we looked at predicting one week ahead using the previous week's data as well as one month ahead using the previous month's data. It was clear to see that using less data and predicting less ahead produced a better fitting model to the actual data. Despite this, there was an increase in the significantly high values for bitcoin.

Using the prices predicted and the drift parameters, we calculated where we should invest, what we should sell and what we should hold on to on each trading day. This was done through predicting the proportions of the investments based upon previous trends. Hence, if the investment was beneficial after the presence of the transaction was considered, a transaction would be made. This model allows us to keep up with the changing trends which is so common in stocks and not fall behind the market.

At the end of the model we would calculate the total value of our portfolio. A value to compare our final value to would be $74590, which is the amount we would have accumulated having invested entirely into bitcoin and sold at the end of the period.

After running the model, the total value of the portfolio at the end of the period is $34820 which is a significant improvement upon our initial investment. However, we would have received a larger return having invested in bitcoin at the start of the period and held on to it until the end of the period. Although investing in gold lowers our final value, it also lowers our risk in the investment as bitcoin being so volatile could easily crash.

Memorandum - Note to Stakeholders

Market trading has been increasingly common practice amongst many enthusiasts and aspiring investors. Bitcoin and gold, in particular, have significantly contributed towards the market and have seen a very strong positive growth in its market value since its early days. Due to the volatility of cryptocurrency and gold in the shared market, it can be challenging to determine if one should buy, sell or hold onto their investments with these two commodities.

This report will aim to develop a model using past market prices to determine the optimal investment strategy to maximum one's assets.

By using the geometric Brownian motion (GBM) to model future market values, one is able to calculate the drift (average rate of change in the market), to see trends within the current market. The GBM at times will exaggerate these trends and the exaggerations is indicative that the market is increasing very rapidly using that period. During this time, we recommend that investors should buy and allocate as much of their money into into the respective commodity, where the GBM model shows and highlights this exaggerations. An example of this can be seen in Figure 1.

By extension, a portfolio value can be modelled to showcase the expected value of the portfolio per trading day using the Price Forecasting model to predict future prices and the Portfolio Re-balancing Model to distribute funds most beneficially. This will remedy some of the uncertainties associated with the small random deviations found in the GBM model. However, over a long period of time, these deviations do not massively contribute towards out main results.

We may take this one step further by also considering how much one should invest/sell during the peaks and troughs of the market. By factoring in the ideal optimal holding proportion as shown in Eq. (5.7), as well as the transaction model, one would be able to factor in if each transaction (either buying or selling) is going to be profitable towards the portfolio. Ideally, we would like the relationship to be $V_f - V_c > (1+\epsilon)F_t$, where V_f is the future value of the market, V_c is the current value, and F_t is the total fee associated with making this transaction. The error team ϵ is purely there to add randomness and is also to ensure that the inequality is held much stronger when it is true. That is to say that when the inequality is true, we should expect a reasonable profit!

Contents

1 Introduction

Market trading has been increasingly common practice amongst many enthusiasts and aspiring investors. Bitcoin and gold, in particular, have significantly contributed towards the market and have seen a very strong positive growth in its market value since its early days. Due to the volatility of cryptocurrency and gold in the shared market, it can be challenging to determine if one should buy, sell or hold onto their investments with these two commodities.

This report will aim to develop a model using past market prices to determine the optimal investment strategy to maximum one's assets.

The first model optimises the trading strategy based on past market values, and explores how the initial investment has changed over time using this model. Justifications will be made to argue why this model provides the most mathematically optimal strategy. Sensitive analysis will be investigated to explore the effects said optimal strategy has on transaction costs.

2 Assumptions

- **Gold and bitcoin can be bought and sold on a continuous scale.** The smallest unit of bitcoin (or any other cryptocurrency) one can trade is the satoshi, which is 100 millionth of a bitcoin (FrankenField, 2022). Hence, continuity is a reasonable assumption as such small amount of bitcoin will rarely make a difference in the overall value of the portfolio. Even at its maximum value of about \$60000, a satoshi will only be worth \$0.0006. On the other hand, there does not seem to be a concensus on the smallest unit of tradable gold. Different Some trading firms cite the smallest tradable unit of gold is 1 gram (0.03527 ounces) while others cite 0.05 ounces (BullionByPost, n.d.; KJC Bullion, 2022). Since, there is no concensus, we decide to not account for discreteness of the tradable gold. Instead, we will ensure that the model produces meaningful values when rounded to such discrete values, so that applying discreteness later is feasible.
- **Daily investment decisions are based only on the price data up to that day.** We assume that no further information from sources other than the previous data (such as news of future events or price data of other similar assets) are available to the trader. As in such cases, a better model accounting for such additional information can be constructed.
- **Missing prices for any given day represents the market being closed on that day.** Gold data for some of the days, seemingly Christmas and new Year's Day, are not provided. We assume that markets are closed on those days and will treat them the same as other days the market is closed.
- **Given prices for bitcoin and gold are constant throughout the day (00:00 - 23:59).** We are provided with one data point per day. Interpolating values between those data points may not lead to accurate data. Furthermore, stock prices are volatile and constantly change with time. When modelling over a sufficiently long period of time, fluctuations in prices within the day will be minuscule.
- **Cash not invested into either bitcoin or gold does not earn interest and hence, its value will remain constant.** Interest rates are not provided with the data which leads us to assume that the trader is unaware of them. Interest rates are extremely informative in the financial world as they provide a baseline for the market. Many financial models are based on interest

rates such as the Black-Scholes model for option pricing Kenton (2019). Hence, this is also in line with our second assumption.

- **All transactions are valid and occur within the day. i.e. transactions are not recalled/rejected.** Sometimes, transactions can fail for a myriad of reasons (such as identification failure, unusual activity detected in your account, etc) or they can be delayed for such reasons (Webull, n.d.). However, such occurrences are rare and modifying the model to accommodate such cases have little utility.
- **Assets cannot be traded for each other. First, they must be liquidated into cash and then, that cash could be used to buy assets.** Although some financial firms may allow to trade between assets (such as paying in bitcoin), it is not common practice (Walsh, 2021). The general standard is to liquidate the the asset first in order to trade it with another asset.

3 Price Forecasting Model

An essential first step to optimise our returns is to predict asset prices in the future from previous data. Once we have a decent prediction of the future, we can use that value to aid us in our investment choices.

There are many approaches one could take to forecast the future prices, all of which have their advantages and disadvantages. We have chosen to use geometric Brownian motions which take into account long term trends, as well as short term randomness.

3.1 Geometric Brownian Motion (GBM)

Stocks are known to be risky investments due to the ever changing (and usually unforeseeable) drop in their market value. In this report, Geometric Brownian Motion (GBM) will be used to model future market values of bitcoin and gold by utilising past data of those respective stock prices each day. The GBM model assumes a constant volatility; however (as we shall later see), it will fail to account for large changes in price.

The GBM model requires inputs of the expected percentage return, standard deviation of the percentage returns, the current price, the number of trading days and the number of days desired to predict for. This means that the growth of the asset could be modelled with as little as three days worth of data (Yildiz, 2019).

The price of an asset $p(t)$, at any given time t, is govern by

$$p(t+1) = p(t) \times \exp\left((\mu - \frac{\sigma^2}{2})dt + \sigma\epsilon\sqrt{dt}\right), \tag{3.1}$$

where dt is the reciprocal of the number of trading days in the year for the asset, ϵ is a randomly generated value from the standard normal distribution with mean 0 and standard deviation 1; $\sigma = sd \times \sqrt{1/dt}$, whereby, sd is the standard deviation of the differences in prices (expressed as a percentage) between consecutive days of the asset $\mu = (E[d] + 1)^{1/dt} - 1$. Further, d is the differences in prices (expressed as a percentage) between consecutive days of the asset. Note, μ and σ have been annualised to be of the same scale to the number of trading days in a year for gold (252) and bitcoin (365) (ScienceDirect, n.d.; Wikipedia, 2020).

Using the GBM model, we were able to predict the future prices based on previous prices and compare them to the actual prices using a line plot.

For each asset, the price has been predicted for each trading day through the period 11/9/2016 - 10/9/2021. Due to the required inputs, at least three days of previous data is required to make a prediction. Additionally, the gold markets are closed on the first day. Therefore, starting from day 4, we predicted the price in the same number of days time that we had previous data for. When we had gathered a month of previous data, we predicted the price in 30 days time using only the previous 30 days of data. As we came within 30 days of the end of the period, we predicted the price until the end of the period and used the same number of days of previous data to do so.

The price on the final day of each prediction was recorded and a graph of those prices can be seen below.

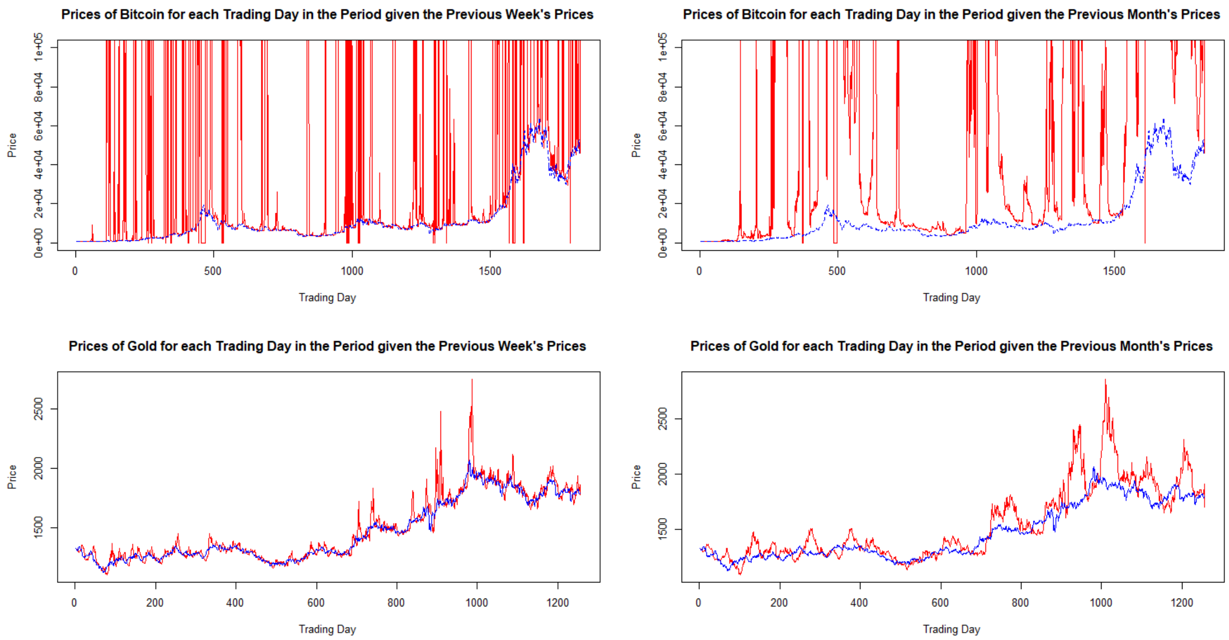

Figure 1: Line plots of the predicted and actual prices for each gold and bitcoin using the previous week's data and the previous month's data. Y axis limit has been set to 100,000 (Morast, 2020).

Modelling the price of the asset given the previous week's prices gives a closer fit to the actual prices in comparison to the model given the previous month's prices. The weekly prediction data seems to fit the actual data's shape more closely. However, it also seems to have more extreme values which may contribute to large amounts of error. Hence, the monthly data was chosen to model future prices.

The GBM model essentially looks to see if the price has increased, decreased or held constant through the previous days. Based on that, it will predict the price for the next days. Therefore, if the price has increased significantly in the past days, it will be predicted that it will continue on that trend. This is particularly evident in the bitcoin models as there are many extreme values due the price increasing by thousands of dollars over small time periods. This is less so evident in the gold models as whilst the prices still peak, the difference is in the hundreds of dollars rather than the thousands.

4 Portfolio Rebalancing Model

Next, we wish to explore a model to optimise our current portfolio with the intention of maximising the value of the assets in the portfolio while minimising the trader's risk.

How shall we approach this? Well, we could look at a similar problem that already has a solution. An index fund is type of financial vehicle made up of a pool of money collected from many investors (Hayes, 2022). This money is invested into a large number of assets to diversify the investment. The idea is to make sure that even if the value of some assets is decreasing, as long as most others' are increasing, the overall portfolio is increasing. By diversifying the investment, index funds reduce "risk". They calculate an index to invest for each asset to optimise future returns. The index depends on the assets' price among other factors which essentially determine the assets' performance in the financial market compared to others (Catalano, 2022). Depending on the asset's index, a proportion of the total funds is invested into the asset. This proportion is re-evaluate frequently to rebalance the distribution of funds. We attempt to do something similar here.

We begin by splitting this portfolio model into the optimisation model, and a transaction model. At the end of the model, the intention is to determine an ideal optimal holding proportion that would indicate how much one should reinvest, sell, or hold in order to become profitable.

4.1 Optimisation Model

To begin the construction of the model, let there be a score model for assets S, such that this score is directly proportional to the rate at which the market will be changing over time. That is to say that

$$S = |d(t)|d'(t), \tag{4.2}$$

where $d(t)$ is known as the drift, and $d'(t)$ is the rate of change in the drift. The drift is the predicted trajectory of asset price corresponding to the average slope of the price, and $d'(t)$ is the drift rate (derivative of the drift). This score ensures that we are always selling and buying the assets close to their turning points to yield maximum profit.

The drift rate $d'(t)$ is calculated using a first order backward finite difference method. Higher order finite differences could be used in an attempt to increase accuracy however, their computation introduces error that counteracts the accuracy and requires more computational power. For our purposes, a first order method is sufficient. Hence, the drift rate is

$$d'(t_i) = \frac{d(t_i) - d(t_{i-1})}{\delta t}, \tag{4.3}$$

where $d'(t_i)$, is the drift rate at time t_i, similarly $d(t_i)$ is the drift at time t_i and δt is the time step which in our case is 1 day. So, the drift rate simplifies to

$$d'(t_i) = d(t_i) - d(t_{i-1}). \tag{4.4}$$

Next, we define the score model for cash to be

$$S_c = -2\frac{S_b S_g}{S_b + S_g}, \tag{4.5}$$

where, S_c, S_b, and $S_g$are the scores for cash, bitcoin and gold respectively. This makes intuitive sense as the score for cash, (which is used to define whether one should invest, or hold onto that case), is dependent on the performance (or scores) of bitcoin and gold. Note that the scalar multiple of two comes from taking the average of the costs of both the bitcoin and gold scores, as shown in the equation above. A quick dimensional analysis also implies that the dimension of S_c is the same as the dimension of S_b and S_g, which is important as in optimal holding proportion model, we add these scores together.

Using the score factor we just defined, we are now motivated to normalised the scores to bound the function in Eq. (7.3) as score that spans from $[-\infty, \infty]$ does not make much physical sense. Let the normalised score

$$N_i = \frac{1}{1 + e^{-S_i}}, \tag{4.6}$$

where N is the normalise score, which maps the asset score S from $[-\infty, \infty]$ to $[0, 1]$ using the sigmoid function, and the index i is used to reference the three commodities of interest here in this report; that is, $i = [c, g, b]$, where c, g, b stands for the usual cash, gold, and bitcoin respectively. For brevity, the index i will be used throughout the report to mean exactly this.

Lastly, we shall define another key parameter P, where P is the optimal holding proportion defined as

$$P_i = \frac{N_i}{N_b + N_g + N_c}, \tag{4.7}$$

where again, the index is used to represent the 3 possible commodities. Note that in the equation above, $\sum_i (P_i) = 1$, as expected. For completeness, P_i may also be thought of as the proportion of funds to be held in the commodity i, and N_c, N_b, N_g are the normalised scores for cash, bitcoin and gold respectively.

4.2 Special Case: Gold Markets Closed

Most markets allow investors to make transaction any time throughout the day. However, these transactions may not necessarily be processed within that day. For the purposes of this report. We will assume that the transaction will occur at the exact instance a transaction is made. For clarity, a transaction will be defined to be the action one performs when buying or selling from the market.

Interestingly, the market for gold and bitcoin and more synonymous. The market for bitcoin is active everyday throughout the week, whilst gold is only active throughout the business week (Monday through to Friday, and the market also closes for gold during Christmas). This change in transactional availability posses a problem with the optimisation model that was previously developed as the optimal holding proportional for all three commodities may not add up to 1 as it previously did before.

To remedy this issue, a condition will be set such that When the gold markets are closed (on Saturday or Sunday), the optimal holding proportion P_g will retain its previous value (gold cannot be traded). In mathematical terms, we wish for $P_{g_{Fri}} = P_{g_{Sat}} = P_{g_{Sun}}$. To avoid multiple indices, the index 'gold' may be dropped in favour of something more meaningful. Further, the calculated optimal holding proportion values of the remaining proportions P_b and P_c can be scaled down so that $P_g + P_b + P_c = 1$.

For the model to be consistent with the market closure time of gold, we expect the optimal proportion of gold to be the same on Friday, Saturday, Sunday and any other day where the market is closed for gold. However, the current model does not allow for the optimal proportion of gold to always be the same during the market closure day due to its dependencies on normalised gold score N_G and thus, depends in the gold score S_g.

Recall that this may be accounted for by simply equating

$$P_{Fri} = \frac{N_g}{N_g + N_b + N_c} = P_{Sat} = P_{Sun} \tag{4.8}$$

Using the normalised scores from Eq. (7.4), $N_i = 1/(1 + e^{-S_i})$, and rearranging this gives

$$S_g = -\ln\left(\frac{1 - P_{Fri}}{P_{Fri}(N_b + N_c)} - 1\right) \tag{4.9}$$

Where $N_c = 1/(1 + e^{-S_c})$, and $S_c = -2(S_b S_g)/(S_b + S_g)$ as usual.

This is a transcendent equation, which requires numerical methods in order to be solved. Solving for S_g will will allow for one to choose a gold score S_g that will be ensure that the optimal holding proportion for any weekend (say Saturday and Sunday), will be the same as the optimal holding proportion that it was on Friday.

As a final note, S_g may also be calculated using machine learning to search for an S_g that will yield the ideal P_g to ensure that $P_{Fri} = P_{Sat} = P_{Sun}$. This may be done by running thousands of simulations and guessing (and checking) the values of P_g to condition it so that $P_{Fri} = P_{Sat} = P_{Sun}$ is true to a some accurate level of certainty.

4.3 Transaction Model

The aim of the transaction model is to determine whether it is more profitable to make a transaction. This may be done by looking at the difference between current proportion held and optimal proportion to hold. By calculating the amount to sell or buy for each asset, and determining the fees and sum them, will allow for the model to check if difference between predicted portfolio value in future with optimal proportions and current proportions is higher than fees (maybe higher than $(1 + \epsilon)$ * sum of fees or something to account for error). The error term ϵ, is defined to be $0 \leq \epsilon \ll 1$, where ϵ takes into account small and random deviations that the Brownian Model is subject to.

Before developing the model, we need to choose an initial investment. The price forecasting model is only valid after 3 days. Therefore, the transaction model which relies on it will also only be valid after 3 days. In such a case, we have three main options. First, we make no blind investments and only make an investment once we have a model, $P_{init} = [1, 0, 0]$. Second, we distribute our funds evenly so that we hold a third of funds as each asset, $P_{init} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$. Third, we trust the asset prices will increase and split our funds two ways to invest in the assets and hold no cash, $P_{init} = [0, \frac{1}{2}, \frac{1}{2}]$. All of these approaches are valid approaches. The first approach is the least risky and the third approach is the most risky because for those 3 days, we have no prediction for the future values of the assets. For purposes of demonstration, we will choose the least risky approach.

We start developing the model by considering the optimal holding proportion P_{opt}, and the current proportion held in our portfolio P_{cur}. Both proportions have elements $[P_c, P_g, P_b]$, where P_c, P_g, P_b are our respective optimal holding proportion for cash, hold and bitcoin. Intuitively, the goal is to determine how one could change the current proportion P_{cur} to the "ideal" and optimal proportion P_{opt}, as the closer P_{cur} is to P_{opt}, the more profitable our portfolio will be.

This motivates one to compute the change in funds held as the asset after transaction

$$G_c = (P_{opt} - P_{cur})G_h \tag{4.10}$$

where G_h is the funds currently held as gold and G_c is the change in gold funds required for the transaction.

Taking this further to include the transaction costs yields $|G_c| * \alpha_g = F_g$, where F_g, would be the fee for gold. The same mathematics can be done for bitcoin and the total fee for the transaction would simply be

$$F_t = F_b + F_g \tag{4.11}$$

Next, the Brownian model can be used to extrapolate future data, which when factored in with the transaction model gives

Explain how V is calculated (portfolio value)

$$V_f - V_c > (1 + \epsilon)F_t \tag{4.12}$$

where ϵ as previously mentioned is defined to be the small random error to account for the randomness in the Brownian Model. Note that the equation above states that if the difference in the value of the future price V_f, and current price V_c is greater than the total transaction fee F_t (whilst also taking into account some randomness), than it would be profitable to make The transaction to move the current proportion P_{cur} to the optimal proportion P_{opt}.

5 Results

As can be seen in Figure 2, the portfolio's value seems to increase overall when using the Price Forecasting and Portfolio Rebalancing models. It does decrease at certain points however, that is likely due to the value of both assets decreasing at those times. Upon closer inspection, the keen-eyed, may have identified that the total portfolio values are reminiscent of the bitcoin values.

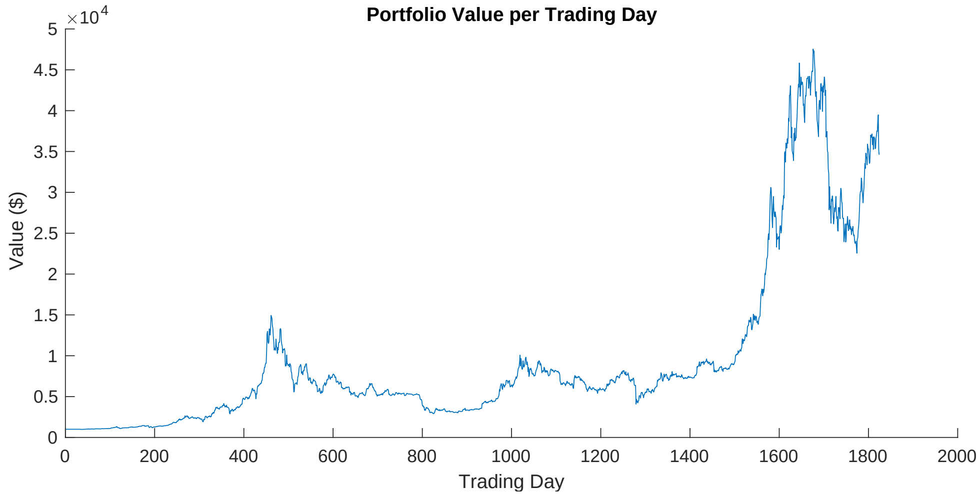

Figure 2: The total value of the portfolio on each trading day using the Price Forecasting model to predict future prices and the Portfolio Rebalancing Model to distribute funds most beneficially.

Let us explore this a bit more by plotting the values of bitcoin, gold, and the portfolio values on the same axes. As Figure 3 shows, the curve of the portfolio value appears to merely be a scaled down version of the curve for bitcoin. Why might this be? Let us explore a bit further.

In Figure 3, the gold value is seemingly constant and close to zero. Well, this can be explained by the scaling factor at the top left corner of the graph which is easily missed. The gold value is substantially large and is changing often as well. However, compared to the value of bitcoin, gold value may as well be constant. This indicates that our bitcoin holdings account for a great portion of our portfolio value.

Large changes in value are characteristic of large volatility. Bitcoin (and cryptocurrencies in general) tend to have substantially larger volatilities than other assets due to the alignment of factors such as supply and demand, investor and user sentiments, government regulations, and media hype (Lapin, 2021; **?**). Furthermore, as we saw in Section 3.1, since bitcoin is traded on more days in a year, its volatility is multiplied by a larger number (365) than gold (252) when annualising.

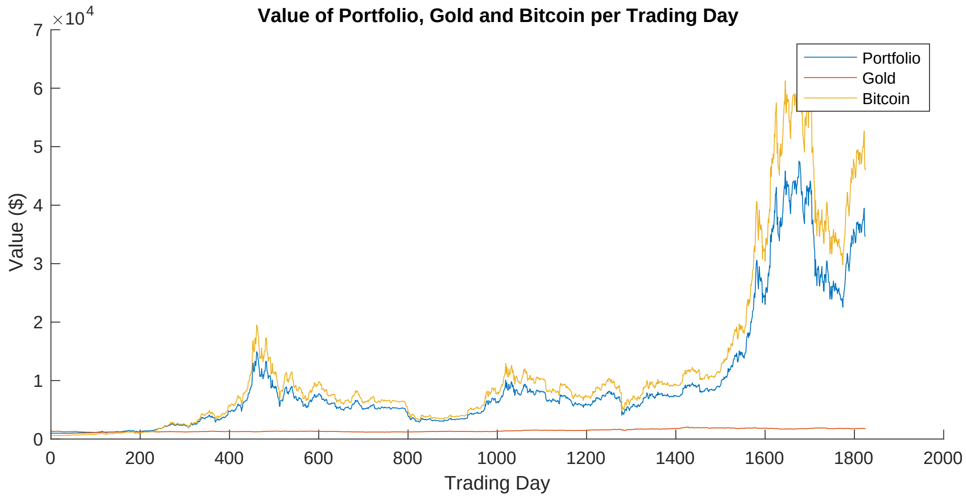

Figure 3: The values of bitcoin, gold and the portfolio plotted on the same axes for the sake of comparison.

The Larger volatility makes bitcoin a "riskier" investment however, as Thomas Jefferson once famously stated, “With great risk comes great reward.” The smaller changes in gold value will never be as profitable bitcoin when it is increasing. Investing in bitcoin can be extremely profitable but it could also be extremely loss-making. This is exactly why we have another asset to invest in —gold. As we discussed in Section 4, the benefit of diversifying your investment is that it reduces the risk.

That is essentially what our model appears to do. It tries to get as much "rewards" as possible, by "risking" as least as possible. This is why we see that when the bitcoin value was changing less drastically, our model was much closer to its value but when it was changing rapidly, there is greater difference between bitcoin and portfolio values. Overall, due to out funds being diversified, our portfolio value is somewhat of a weighted average between the values of the assets we are modelling.

If one could predict with certainty from the first day that bitcoin was going to explode, then it would be a great idea to put all of our eggs in one basket. If not, then it is better to be safe than sorry because it was possible for bitcoin to complete tank. At that point, we'd probably be thanking our old friend gold whom we were just condemning for weighing us down.

P_{init}	Final Portfolio Value
$[1,0,0]$	\$34820
$[\frac{1}{3},\frac{1}{3},\frac{1}{3}]$	\$35113
$[0,\frac{1}{2},\frac{1}{2}]$	\$38710

Table 1: Final portfolio values depending on initial investment proportions.

Now, one may be wondering, how much does the trader end up making if they use this model. The answer can be seen in Table 1. The final portfolio values are all of the same order of magnitude, regardless of starting proportion. The starting proportion did have an effect on the final portfolio values, but the effect was small indicating that even though some starting proportions were more risky than others, that difference is not substantial.

6 Strengths and Weaknesses

No model is perfect and all have their strengths and weaknesses. Our model resulted in us significantly improving on our investment, however, there was potential to earn a much higher return.

Our model predicted the prices based on the drift using the previous days data well. The predictions followed the trend of the actual prices ignoring the significantly large fluctuations in price. Therefore, we could tell whether the price is likely to increase or decrease in the next days. As we decreased the length of time we predict for and the amount of previous data used, the predictions would follow the trend of the actual values better but it caused more significantly high price fluctuations.

These fluctuations were more common in the bitcoin data due to the higher risk higher reward of the asset. Therefore, whilst our model fit the data for gold well, it was lacking when predicting an asset with a higher value and with larger jumps in prices. Due to the significantly large fluctuations caused by the large jumps in the bitcoin prices, we may have overestimated how much to invest. Consequently, we would sell at a point where we predicted would be extremely high but in reality was no where near that price.

7 Conclusions

Stocks being as volatile as they are, are always hard to predict and result in winners and many losers. Looking at the path of the value of gold and bitcoin from 11/9/2016 - 10/8/2021, we aimed to maximise the value of our portfolio at the end of the period having had \$1000 in cash at the beginning of the period. On each trading day, we decided to buy, sell or hold, after considering the transaction for gold (1%) and bitcoin (2%).

We began with a geometric brownian motion (GBM) model to calculate the drift and the price of the assets on future days. Due to the high risk - high reward property that was more so present in bitcoin, the model overestimated it's growth at times resulting in significantly large prices. This wasn't present in gold due to its lower price and hence lower reward.

Using the prices predicted and the drift parameters, we calculated when to buy, sell and hold. This began by predicting the proportions in which each we should invest in each asset based upon previous trends. After accounting for the transaction fee, then deciding whether to go ahead with that transaction.

Our model predicted that the final value of our portfolio is \$34820. This is a significant growth on our initial investment but still less that the amount we would have accumulated having invested in bitcoin at the start and held onto it until the end of the period.

A Appendix

All of the code developed for this project is accessible at `https://github.com/TheSurgeonOfDeath/MCM`. However, the most important scripts and functions are displayed below.

A.1 Price Forecasting Model

```
pacman::p_load(tidyr, tidyverse)

bitcoin <- read.csv("bitcoin.csv")


# Data Handling Function
    ↪ --------------------------------------------------

summary_data <- function(data){
  # input data and remove missing data
  summary <- data
  summary <- na.omit(summary)

  # define zero matrices
  exp_price <- matrix(NA, nrow = length(summary$Value), ncol = 1)
  sd <- matrix(NA, nrow = length(summary$Value), ncol = 1)

  # finding running mean and standard deviation
  for (i in 1:length(summary$Value)){
    run <- summary[1:i,2]
    exp_price [i,1] <- mean(run, na.rm = TRUE)
    sd[i,1] <- sd(run, na.rm = TRUE)
  }
  # add to data frame
  summary <- data.frame(summary, exp_price, sd)

  # define zero matrices
  diff <- matrix(NA, nrow = length(summary$Value), ncol = 1)
  perc_diff <- matrix(NA, nrow = length(summary$Value), ncol = 1)

  # calculate differences and percentage differences
  for (i in 2:length(summary$Value)-1){
    if (!is.nan(summary[i,2])){
      x<- summary[i,2]
    }
    if (!is.nan(summary[i+1,2])){
      diff[i+1] <- (summary[i+1,2] - x)
      perc_diff[i+1] <- (summary[i+1,2]-x)/x
    }
  }
```

```
  # add to data frame
  summary <- data.frame(summary, diff, perc_diff)

  # define zero matrices
  exp_perc_change <- matrix(NA, nrow = length(summary$Value), ncol = 1)
  sd_perc_change <- matrix(NA, nrow = length(summary$Value), ncol = 1)

  exp_diff_change <- matrix(NA, nrow = length(summary$Value), ncol = 1)
  sd_diff_change <- matrix(NA, nrow = length(summary$Value), ncol = 1)

  # calculate summary statistics of differences and percentage differences
  for (i in 2:length(summary$Value)){

    run <- summary[1:i,5]

    runa <- summary[1:i,6]

    exp_perc_change [i,1] <- mean(runa, na.rm = TRUE)

    sd_perc_change[i,1] <- sd(runa, na.rm = TRUE)

    exp_diff_change [i,1] <- mean(run, na.rm = TRUE)

    sd_diff_change[i,1] <- sd(run, na.rm = TRUE)

  }

  # add to data frame
  summary <- data.frame(summary, exp_perc_change, sd_perc_change, exp_diff_
     ↪ change, sd_diff_change)


  return(summary)
}



# Call Data Handling Function
   ↪ -------------------------------------------------------


summary <- summary_data(bitcoin)




# Price Predictions GBM
   ↪ --------------------------------------------------------
```

```
gbm_price_pred <- function(data, nsim = 100, t = 25, S0 = 100, td = 252,
    ↪ current_day = 10) {
  # adjustment for t ( code functionality )
  t <- t + 1

  b <- current_day - t
  if (b<1){
    b = 1
  }
  runa <- data[b:current_day,6]

  exp_perc <- mean(runa, na.rm = TRUE)

  sd_perc <- sd(runa, na.rm = TRUE)
  # calculate drift
  mu <- (exp_perc +1)^td-1
  # calculate volatility
  sigma <- sd_perc * sqrt(td)
  # define matrix
  gbm <- matrix(ncol = nsim, nrow = t)
  # reciprocal of the number of trading days
  dt <- 1/td
  # calculate the predicted price for future days
  set.seed(100)
  e_vec <- rnorm(nsim)
  for (i in 1:nsim) {
    gbm[1, i] <- S0
    for (day in 2:t) {
      e <- e_vec[i]
      gbm[day, i] <- gbm[(day-1), i] * exp((mu - sigma * sigma / 2) * dt +
        ↪ sigma * e * sqrt(dt))
    }
  }

  gbm_df <- as.data.frame(gbm)

  avg <- matrix(0, nrow = dim(gbm_df)[1], ncol = 1)

  avg <- data.frame(avg)

  for (i in 1:dim(gbm_df)[1]){

    run <- gbm_df[i,1:dim(gbm_df)[2]]
    sum <- 0
    for (j in 1:length(run)){
      sum <- sum + run[j]
    }
```

```
    avg[i,1] <- sum/length(run)
  }

  days <- 1:nrow(gbm) + current_day -1
  gbm_df <- data.frame(days, gbm_df, avg)
  end <- current_day + t - 1
  gbm_df <- data.frame(gbm_df, actual=summary$Value[current_day:end])

  # gbm_df <- as.data.frame(gbm) %>%
  # mutate(day = 1:nrow(gbm) + current_day -1) %>%
  # pivot_longer(-day, names_to = 'sim', values_to = 'predicted')
  # end <- current_day + t - 1
  # gbm_df <- data.frame(gbm_df, actual=data$Value[current_day:end])
  # gbm_df <- gbm_df %>%
  # select(day, predicted, actual) %>%
  # gather(key = "variable", value = "value", -day)

  # returning drift and final price
  drift <- mu

  final_price <- gbm_df$avg[dim(gbm_df)[1]]

  final_outputs <- data.frame(drift, final_price)

  return(final_outputs)
}



# Call Price Predictions GBM Function
   ↪ ---------------------------------------------------

gbm_df <- gbm_price_pred(summary, 100, 30, summary$Value[125], 252, 125)

final2 <- gbm_price_pred(summary, 100, 3, summary$Value[3], 252, 3)

for (i in 5:length(summary$Value)-1){
  if (i<30){
    j <- i
  }else if (i>length(summary$Value)-30){
    j<- length(summary$Value)-i
  }else{
    j <- 30
  }
  gbm_df <- gbm_price_pred(summary, 100, j, summary$Value[i], 252, i)

  final2 <- rbind(final2, gbm_df)
```

```
}

present_day <- days <- 1:dim(final2)[1] + 2
y <- data.frame(present_day, final2)

future_day <- matrix(NA, nrow = dim(final2)[1], ncol = 1)

for (i in 4:length(summary$Value)-1){
  if (i<30){
    j <- i
  }else if (i>length(summary$Value)-30){
    j<- length(summary$Value)-i
  }else{
    j <- 30
  }

  future_day[i-2,1] <- i+j
}

y <- data.frame(y, future_day)

write.csv(y,"bitcoin_path.csv", row.names = FALSE)

# Graph Data
   ↪ --------------------------------------------------------------

ggplot(gbm_df, aes(x = day, y = value)) +
  geom_line(aes(color = variable, linetype = variable)) +
  scale_color_manual(values = c("darkred", "steelblue"))+
  theme(plot.title = element_text(size=12, face="bold",
                                  margin = margin(10, 0, 10, 0))) +
  ggtitle("Brownian␣Motion␣Model␣for␣Gold␣Price␣for␣30␣days␣using␣previous␣
     ↪ 30␣day␣data")
```

A.2 Optimisation Model

```
function [Pc, Pg, Pb] = Optimal_Proportions(Sg, Sb)
% Prepare data
% [Sg, Sb] = Prepare_data();

% Normalised bitcoin score (independent of Sg)
Nb = Normalised_score(Sb);

% Calculate Optimal Proportions
Ng = zeros(size(Sg));
Sc = zeros(size(Sg));
Nc = zeros(size(Sg));
Pg = zeros(size(Sg));
```

```
Pb = zeros(size(Sg));
Pc = zeros(size(Sg));

for i = 1:length(Sg)
    % market closed
    if isnan(Sg(i))
        % Calculate gold score when market closed to keep propotion same
        % day before market closed
        j = i;
        while isnan(Sg(j))
            j = j-1;
        end

        % Proportion of gold cannot change
        Pg(i) = Pg(j);

        % Find Sg for day such that Pg does not change (numerically) -
        % guess new score is close to old score.
        disp(i)
        epsilon = max(Sg) - min(Sg);
        Sg_guess = Sg(j);
        if abs(Sg(j)) < 1E-7
            Sg_guess = 1E-7;
        end
        Sg(i) = Sg_numerical(Pg(i), Sb(i), Sg_guess, epsilon);
    end

    % cash score
    Sc(i) = -2 * Sg(i) * Sb(i) / (Sg(i) + Sb(i));

    % normalised score
    Ng(i) = Normalised_score(Sg(i));
    Nc(i) = Normalised_score(Sc(i));

    % optimal proportions
    scoreSum = Ng(i) + Nb(i) + Nc(i);
    Pg(i) = Ng(i) / scoreSum;
    Pb(i) = Nb(i) / scoreSum;
    Pc(i) = Nc(i) / scoreSum;
end
end
```

A.3 Transaction Model

```
% Initial Funds
F_init = [1000, 0, 0];
P_init = F_init./sum(F_init);
V_init = sum(F_init);
```

```
% Day 4 model starts

% read data
bitcoin_path = import_file_as_matrix(pwd + "/Final/
    ↪ bitcoin_monthly_annualised.csv");
gold_path = import_file_as_matrix(pwd + "/Final/gold_monthly_annualised.
    ↪ csv");
gold = readtable(pwd + "/../gold.csv");
bitcoin = readtable(pwd + "/../bitcoin.csv");

% Prelimary gold and bitcoin scores (cleaned)
[Sg, Sb] = Prelim_scores(gold_path, gold, bitcoin_path);

% Optimal proportions
% [Pc, Pg, Pb] = Optimal_Proportions(Sg, Sb);
% P_opt = [Pc, Pg, Pb];
% P_opt = [zeros(2,3); P_opt]; % model starts at day 4
load('P_opt_monthly.mat')

% Current proportions
P_cur = zeros(size(P_opt));

% Actual prices
priceB_cur = table2array(bitcoin(2:end,2));
priceG_cur = table2array(gold(2:end,2));
for i = 1:length(priceG_cur)
    if isnan(priceG_cur(i))
        priceG_cur(i) = priceG_cur(i-1);
    end
end
A_price_cur = [ones(size(priceB_cur)), priceB_cur, priceG_cur];

% Predicted future prices
priceB_fut = bitcoin_path(:,3);
priceG_fut = gold_path(:,3);
% model start on day 4 so assume value of stock stays constant in future
% for first 3 days.
priceB_fut = [priceB_cur(1:2); priceB_fut];
priceG_fut = [priceG_cur(1:2); priceG_fut];
[priceG_fut, priceB_fut] = Correct_price_dimensions(priceG_fut,priceB_fut)
    ↪ ;

% Remove infinity
priceB_fut = Remove_extremes(priceB_fut, 2);
priceG_fut = Remove_extremes(priceG_fut, 2);
for i = 1:length(priceG_fut)
    if isnan(priceG_fut(i))
```

```
        priceG_fut(i) = priceG_fut(i-1);
    end
end

A_price_fut = [ones(size(priceB_fut)), priceB_fut, priceG_fut];


% Total funds
F = zeros(size(P_opt));

% Asset amounts
A_amount = zeros(size(P_opt));




% Investment on Day 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% P_invest = P_init; % [1, 0, 0] no blind investment
P_invest = [0, 0.5, 0.5]; % 2 way split
% P_invest = [1/3, 1/3, 1/3]; % 3 way split
F_invest = P_invest .* sum(F_init);
F(1,:) = F_invest;
A_amount(1,:) = F_invest ./ A_price_cur(1,:);
P_opt(1,:) = P_invest;
for i = 2:4
    A_amount(i,:) = A_amount(i-1,:);
    F(i,:) = A_amount(i,:) .* A_price_cur(i-1,:);
    P_opt(i,:) = P_invest;
end
P_cur(1:3,:) = P_opt(1:3,:);




% Fee percent
alpha = [0,0.01,0.02];

% Error
epsilon = 0;


% Transactions
num_trans = 0;
% [C, G, B]
for i = 3:length(A_price_fut)
    % Asset amount today
    A_amount(i,:) = A_amount(i-1,:);
```

```
    % Funds today
    F(i,:) = A_amount(i, :) .* A_price_cur(i,:);

    % Current proportion
    P_cur(i,:) = F(i,:) ./ sum(F(i,:));

    % Asset change
    Ac = (P_opt(i, :) - P_cur(i, :)) .* sum(F(i,:));

    % Fee on transactions
    fee = alpha .* abs(Ac);
    feeTot = sum(fee);

    % total current funds before and after transaction
    Vc_before = sum(F(i,:));
    Vc_after = Vc_before - feeTot;

    % asset amounts after transaction
    F_after = P_opt(i, :) .* Vc_after;
    A_amount_after = F_after ./ A_price_cur(i,:);

    % asset value in future
    A_val_fut_before = A_amount(i,:) .* A_price_fut(i,:);
    A_val_fut_after = A_amount_after .* A_price_fut(i,:);

    % Future Portfolio value before and after transaction
    Vf_before = sum(A_val_fut_before);
    Vf_after = sum(A_val_fut_after);

    % Make transaction?
    if(Vf_after - Vf_before > (1+epsilon)*feeTot)
        num_trans = num_trans + 1;
        P_cur(i,:) = P_opt(i,:);

        % new funds today after transaction
        F(i,:) = Vc_after .* P_opt(i,:);
        A_amount(i,:) = F(i,:) ./ A_price_cur(i,:);
    end
end



% Final portfolio value
% (previous amount * current price) - amount vector is shorter by 1 day
F_final = A_amount(end,:) .* A_price_cur(end, :);
V_final = sum(F_final);
disp(V_final);
```

```
% Comparison by investing all in bitcoin
b_price = table2array(bitcoin(:,2));
b_amount_init = 1000 / b_price(1);
b_peak_price = max(b_price);
b_end_price = b_price(end);
b_value_end = b_amount_init * b_peak_price;
b_value_end2 = b_amount_init * b_end_price;

diff = V_final - b_value_end;
diff2 = V_final - b_value_end2;
disp(diff);
disp(diff2);

% Plots
hold on;
V = sum(F,2);
plot(V, 'DisplayName', 'Portfolio');
plot(priceG_cur, 'DisplayName', 'Gold')
plot(priceB_cur, 'DisplayName', 'Bitcoin')
title("Gold Value per Trading Day")
xlabel("Trading Day")
ylabel("Value ($)")
legend;
hold off;
```

References

BullionByPost. (n.d.). *Minimum amount to invest in gold? | bullionbypost.* Retrieved 2022-02-21, from https://www.bullionbypost.co.uk/index/gold/minimum-amount-to-invest-in-gold/

Catalano, T. J. (2022). *What the dow means and how it is calculated.* Retrieved 2022-02-21, from https://www.investopedia.com/articles/investing/082714/what-dow-means-and-why-we-calculate-it-way-we-do.asp#:~:text=The%20index%20is%20calculated%20by

FrankenField, J. (2022). *Satoshi.* Retrieved 2022-02-21, from https://www.investopedia.com/terms/s/satoshi.asp#:~:text=The%20satoshi%20is%20the%20smallest

Hayes, A. (2022). *Mutual fund definition.* Retrieved from https://www.investopedia.com/terms/m/mutualfund.asp#:~:text=A%20mutual%20fund%20is%20a

Kenton, W. (2019). *How the black scholes price model works.* Retrieved from https://www.investopedia.com/terms/b/blackscholes.asp

KJC Bullion. (2022). *Kjc bullion - australia's largest range of gold and silver.* Retrieved 2022-02-21, from https://www.kjc-gold-silver-bullion.com.au/

Lapin, N. (2021). *Explaining crypto's volatility.* Retrieved 2022-02-21, from https://www.forbes.com/sites/nicolelapin/2021/12/23/explaining-cryptos-volatility/?sh=1728416a7b54

Morast, A. (2020, 10). *Simulating stock prices using geometric brownian motion.* Retrieved from https://medium.com/analytics-vidhya/simulating-stock-prices-using-geometric-brownian-motion-f400bb6af6dd

ScienceDirect. (n.d.). *Geometric brownian motion - an overview | sciencedirect topics.* Retrieved from https://www.sciencedirect.com/topics/mathematics/geometric-brownian-motion

Walsh, D. (2021, 12). *The major companies that accept bitcoin and other cryptos as payment.* Retrieved from https://www.euronews.com/next/2021/12/04/paying-with-cryptocurrencies-these-are-the-major-companies-that-accept-cryptos-as-payment

Webull. (n.d.). *How long does the stock transfer take?* Retrieved from https://www.webull.com/help/faq/30-How-long-does-stock-transfer-take

Wikipedia. (2020, 07). *Stochastic drift.* Retrieved from https://en.wikipedia.org/wiki/Stochastic_drift#:~:text=In%20probability%20theory%2C%20stochastic%20drift

Yildiz, U. (2019, 08). *Simulating stock prices in python using geometric brownian motion.* Retrieved from https://towardsdatascience.com/simulating-stock-prices-in-python-using-geometric-brownian-motion-8dfd6e8c6b18