Problem **C**

2022 MCM/ICM Summary Sheet

Team control number **2228316**

An Empirical Distribution Function (EDF) Trading Strategy

Summary

Before we began construction of an algorithm, we first considered each data set as a time series and fit ARIMA models to both sets of data. We conducted Ljung-Box tests on both sets of residuals and determined that each set of residuals were independent.

Our empirical distribution function (EDF) algorithm uses this assumption of independence to calculate EDFs that estimate the probability of obtaining a residual less than or equal to the residual for a given date d_t. A damping factor is introduced to account for the convergence rate of the empirical distribution function[1] and resulting estimate uncertainty at earlier dates. Our algorithm uses the product of the damping factor and EDF probability to calculate the percentage of current assets to sell and percentage of current cash to invest.

Experimental results show that our EDF algorithm is quite successful, producing a final September 10, 2021 portfolio value of 71,562 USD. However, our algorithm is somewhat sensitive to transaction costs, as setting $\alpha_{\text{gold}} = \alpha_{\text{bitcoin}} = 0.1$ produces a final portfolio value of 58187.83 USD and setting $\alpha_{\text{gold}} = \alpha_{\text{bitcoin}} = 0.2$ produces a final portfolio value of 55770.13 USD.

Keywords: EDF, ARIMA, time series, residual

An Empirical Distribution Function (EDF) Trading Strategy

February 21, 2022

Contents

1 Introduction **3**
1.1 Background . . . 3
1.2 Our Work . . . 3
1.3 Data Pre-processing . . . 3

2 Assumptions **7**

3 Abbreviations and Definitions **8**

4 Algorithms **8**
4.1 Random Algorithm . . . 8
4.2 Proportional Percentage Algorithm . . . 9
4.3 Empirical Distribution Function (EDF) Algorithm . . . 10

5 Sensitivity Analysis **11**

6 Strengths and Weaknesses **12**
6.1 Strengths . . . 12
6.2 Weaknesses . . . 12

7 Conclusion **12**

Appendices **13**

Bibliography **13**

1 Introduction

1.1 Background

Market traders buy and sell volatile assets frequently, with a goal to maximize their total return. Determining the best trading strategy for a given asset at each time is thus an important problem in finance.

1.2 Our Work

We aim to develop a model that uses only the past stream of daily prices to date to determine how much of their assets a trader should buy, sell, or hold each day. We assume that each trader's portfolio consists only of cash, gold, and bitcoin in U.S. dollars, troy ounces, and bitcoins, respectively.

Task 1 *We build models that predict the future price of each asset based on only the past stream of daily prices to date.*

Task 2 *From those models, we create an algorithm that determines the best trading strategy for a given asset at each time, again based on only the past stream of daily prices to date.*

1.3 Data Pre-processing

We began by loading the provided CSV files into a Jupyter Notebook as pandas dataframes. These dataframes have two columns: one listing the dates from 9/11/2016 to 9/10/2021, and another listing the market price on the associated date (either the USD value per bitcoin or the USD value per troy ounce of gold, depending on the dataframe). We re-index these dataframes, with the dates as the indices. In the dataframe concerning gold prices, we notice that certain dates have no associated value, indicating that the gold market is not open on these days. In order to account for this, we drop any NaN values from the table containing gold prices. Pictured below, in Figures 1 and 2, are the resulting dataframes and time series.

```
Date
2016-09-12    1324.60
2016-09-13    1323.65
2016-09-14    1321.75
2016-09-15    1310.80
2016-09-16    1308.35
               ...
2021-09-06    1821.60
2021-09-07    1802.15
2021-09-08    1786.00
2021-09-09    1788.25
2021-09-10    1794.60
Name: USD (PM), Length: 1255, dtype: float64
```

Figure 1: Loaded gold data

```
Date
2016-09-11      621.65
2016-09-12      609.67
2016-09-13      610.92
2016-09-14      608.82
2016-09-15      610.38
                 ...
2021-09-06    51769.06
2021-09-07    52677.40
2021-09-08    46809.17
2021-09-09    46078.38
2021-09-10    46368.69
Name: Value, Length: 1826, dtype: float64
```

Figure 2: Loaded bitcoin data

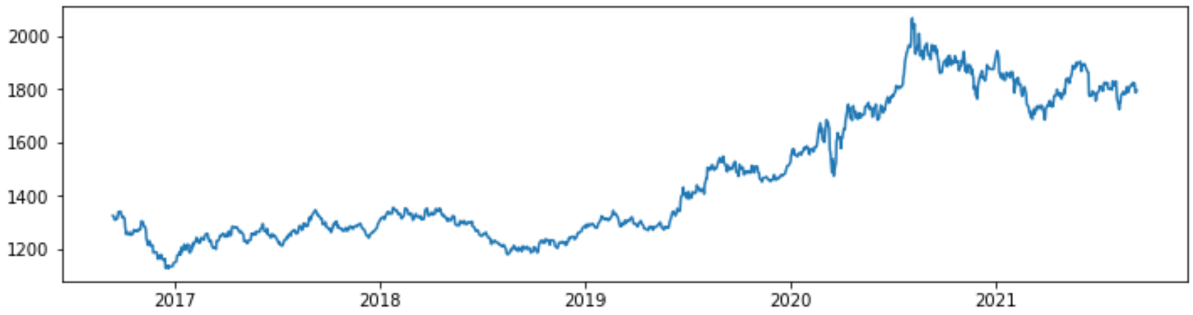

Figure 3: Gold Time Series

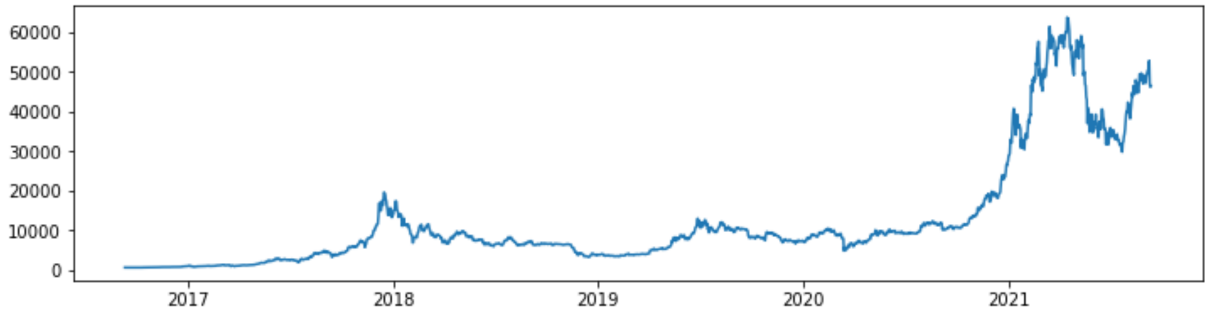

Figure 4: Bitcoin Time Series

We fit ARIMA models to both time series. Results are shown in figures 5 and 6.

```
Series: train_gold.ts
ARIMA(0,1,0)

sigma^2 = 194:  log likelihood = -5033.64
AIC=10069.27   AICc=10069.28   BIC=10074.4
```

Figure 5: Gold ARIMA model

```
Series: train_bitcoin.ts
ARIMA(0,1,2) with drift

Coefficients:
          ma1     ma2    drift
      -0.0804  0.0832  26.6402
s.e.   0.0235  0.0231  18.5864

sigma^2 = 623837:  log likelihood = -14667.06
AIC=29342.12   AICc=29342.14   BIC=29364.13
```

Figure 6: Bitcoin ARIMA model

Running the Ljung-Box test[2] on the gold and bitcoin model residuals, we get the following:

```
        Box-Ljung test

data:  fit_gold$residuals
X-squared = 0.55902, df = 1, p-value = 0.4547
```

```
        Box-Ljung test

data:  fit_bitcoin$residuals
X-squared = 0.0091832, df = 1, p-value = 0.9237
```

Thus, for both models, we fail to reject the null that the residuals are independently distributed. Below are the autocorrelation functions (ACF) and partial autocorrelation functions (PCF) for both gold and bitcoin residuals.

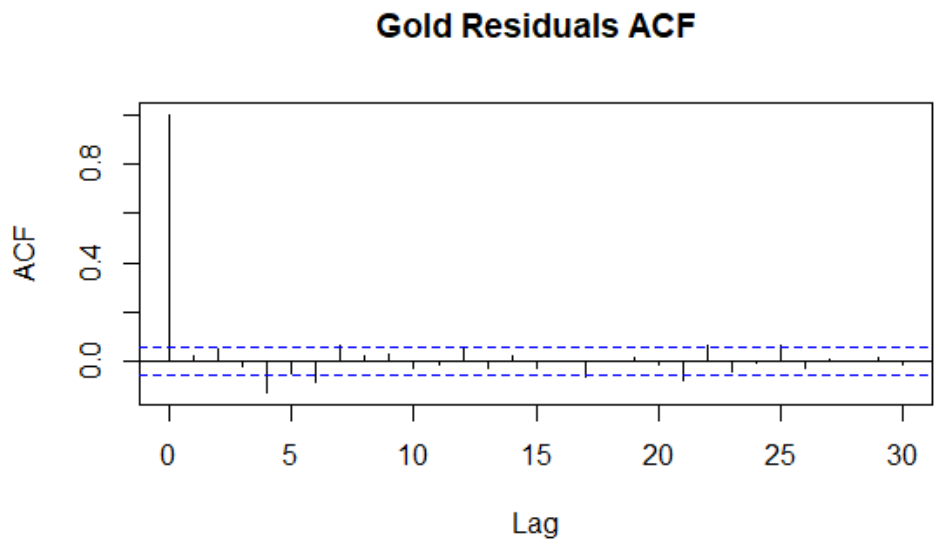

Figure 7: Gold Residuals ACF

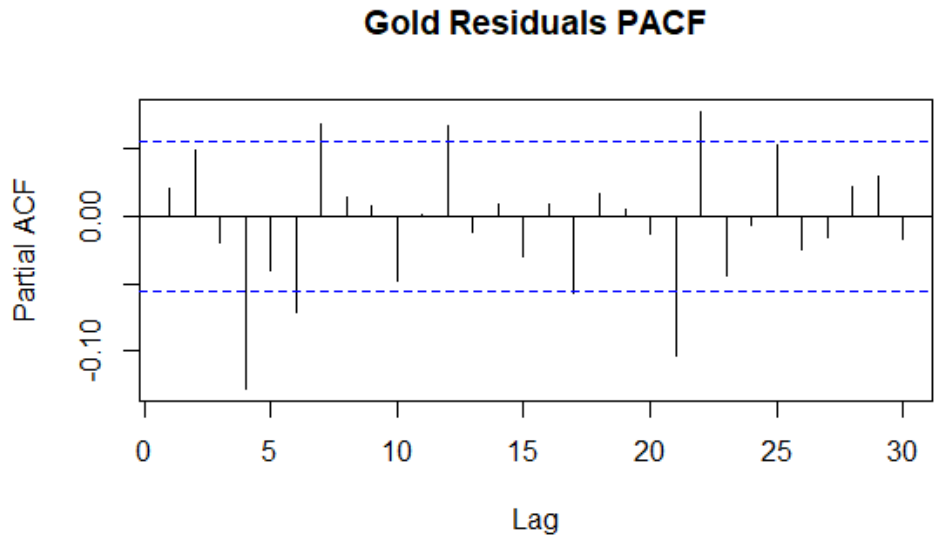

Figure 8: Gold Residuals PACF

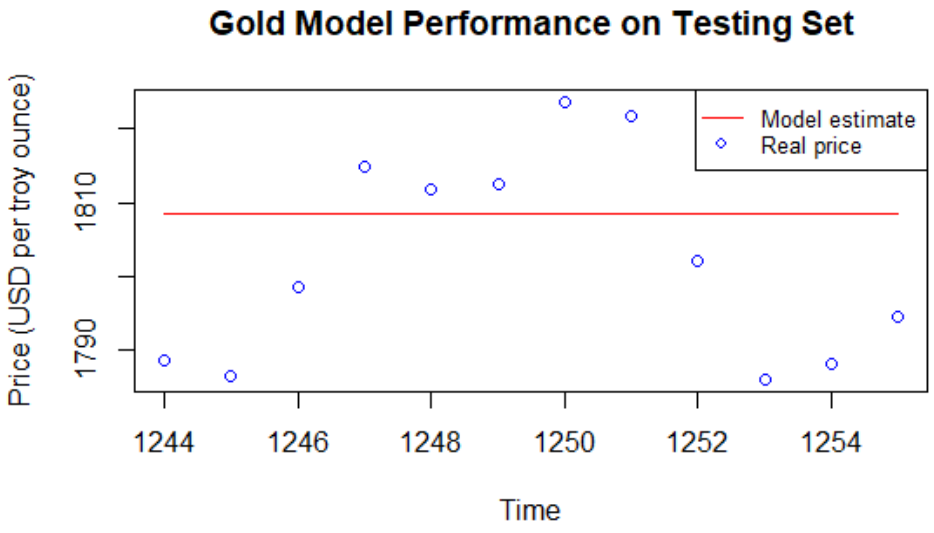

Figure 9: Gold Model Performance on Testing Set

2 Assumptions

Assumption 1: Inflation is zero.
Justification: *For model simplification, we assume there is zero overall rise in the price of goods and services, which eliminates several external factors.*

Assumption 2: The trader cannot have negative cash or assets at any point
Justification: *For model simplification, we assume the trader cannot borrow to finance transactions and thus cannot accrue debt.*

3 Abbreviations and Definitions

Symbol	Definition	Unit
d_g^n	days between September 12, 2016 and the nth date in the gold data set	days
d_b^n	days between September 11, 2016 and the nth date in the bitcoin data set	days
d_n	the nth smallest element of $\{d_g^t\} \cup \{d_b^t\}$	days
s_g^n	the nth residual from the gold ARIMA model	USD/troy ounce
s_b^n	the nth residual from the bitcoin ARIMA model	USD/bitcoin
D_g	total number of days with recorded data for gold	days
D_b	total number of days with recorded data for bitcoin	days
p_g^n	the percentile of s_g^n in the set $\{s_g^1, \ldots, s_g^n\}$	/
p_b^n	the percentile of s_b^n in the set $\{s_b^1, \ldots, s_b^n\}$	/
θ	constant optimized for specific subset of data	/

4 Algorithms

4.1 Random Algorithm

Our first approach, in order to set a benchmark, is to develop a random model in which the trader first buys bitcoin and gold (if the gold market is open on the given date) using a random percentage the USD they currently possess in their portfolio, and then sell a random percentage of their bitcoin (in bitcoins) and then a random percentage of their gold (in troy ounces). We run this model for each day of our given time period, updating the trader's portfolio based on the model's decisions as we go. In the end, we calculate the overall value of the portfolio in USD based on the current gold and bitcoin prices in order to judge how much the initial investment of $1000 is now worth on 09/10/2021. Using this model, we obtain a final value (in USD) of 1.5992×10^{-7} (averaged over 100 trials).

As expected, the random model on its own results in an abysmal final value. We hypothesize that this is likely because the randomness of the percentages we generate to determine how much gold or bitcoin to trade holds no regard for the transaction costs; in order to refine the model, then, we must make our percentages smaller. In our initial random model, the random percentages we used to guide our trading were just random numbers generated from the interval (0, 1); we try dividing this random number by 100 and using that as the new percentage and observe a much better result (over 100 trials, an average final value of 8670.1266 USD).

Our last step in refining this random model is varying the value by which we divide the random number (which is, recall, chosen from the interval (0, 1)) we use in order to determine which percentage of each asset the trader should buy or sell. We loop first through the numbers [1, 10, 100, 1000, 10000] as the divisors of our randomly generated number, and find that the highest final value is generated when that divisor is set to 100. This indicates that our optimal divisor is likely on the order of 100, so we loop through the numbers [100, 200, 300, 400, 500, 600, 700, 800, 900]. From this we discover that the divisor that results in the best final asset value is 200. Our final random model, then, is one in which the percentage

according to which the trader buys or sells their assets is a randomly generated number within (0, 1) divided by 200. Using this random model, we obtain an average final value of 9,474.2366 USD.

We have thus acquired a benchmark value; we know that, through random buying and selling, a trader can end up with 9474.2366 USD, and so we want our actual models to be able to beat this value.

We notice that there is a huge spike in bitcoin prices that occurs near the end of the given time period. In order to account for this and to gain a better understanding of how our random model performs in more consistent conditions, we calculate the final value that results by running our random model on only the first 80% of our data. At the end of the first four-fifths of the period, the optimal random model generates an average final value of 8,892.9234 USD. We now have two benchmarks against which we can compare future models.

4.2 Proportional Percentage Algorithm

Let us now try to eliminate some of the randomness by modifying the way in which we calculate what percentage of each asset to buy or sell. This time, we will use the difference between the current date's price and the last (valid) date's price and make our percentage proportional to this difference.

First, we set up our model so that we sell some percentage of the asset if the aforementioned difference in its price is positive, and buy some percentage of it otherwise. The reasoning here is that if an asset's price has increased since the previous time point, then we will likely be making a profit upon selling it. Likewise, if an asset's price has dropped, then we should buy it now and wait for the price to rise so we can profit.

Now we must calculate the specific percentages of assets we wish to buy or sell. Because we want our percentage to be some number between 0 and 1, we begin by defining the percentage as equal to the current difference divided by the maximum difference that occurs between two subsequent trading dates over the entire time period. Because this difference may be negative, we must also take its absolute value. Using this value as the percentage of the asset that we wish to buy or sell, we find that we end up, on average over 100 trials, with a final value of 1,398.2529 USD. We may have gained money, but this is worse than our best random model.

Similar to what we did with our random model, we attempt to refine this model by changing the percentage calculation. We divide our previous percentage calculations by 10, 100, 1000, and 10000, in hopes that a more conservative trading strategy will prove more fruitful, but our final value actually appears to decrease with each further division of our percentage.

We thus try switching our strategy of when to buy and sell assets. Leaving the percentage calculations the same, we instead attempt make the model such that we buy an asset if the calculated difference between its current price and its previous price is positive, and sell it if the difference negative. Here, we logic that this could work because if an asset's price has gone up, then it will likely continue its upward trajectory for some time. If it does not, then we sell it at the first instance where it appears to decrease, in order to cut our losses.

This results in a much higher final value–an average of 12,544.0585 USD over 100 trials, which is quite a bit better than our random model. In order to get a more measured view of

how the model will act, we again consider only the first 80% of time points and calculate the final value at the end of these. For this, we get a final value of 3,337.9014 USD, which is far less than the value we achieved for the first 80% of the data using the random model.

4.3 Empirical Distribution Function (EDF) Algorithm

Our optimal algorithm is as follows:

Algorithm 1 Best strategy at time d_t

if $d_t \neq d_g^1$ and $d_t \neq d_g^2$ **then**
  if $s_g^t \geq 0$ **then**
    sell $p_g^n \sqrt{\frac{d_n - d_g^1}{D_g}}$ of current gold
  else
    spend $\left(1 - p_g^n\right) \sqrt{\frac{d_n - d_g^1}{D_g}}$ of current cash on gold
  end if
end if
if $d_t \neq d_b^1$ and $d_t \neq d_b^2$ **then**
  if $s_b^t \geq 0$ **then**
    sell $\theta p_b^n \sqrt{\frac{d_n - d_b^1}{D_b}}$ of current bitcoin
  else
    spend $\theta \left(1 - p_b^n\right) \sqrt{\frac{d_n - d_b^1}{D_b}}$ of current cash on bitcoin
  end if
end if

where

- d_g^n is the number of days between September 12, 2016 and the nth date in the gold data set
- d_b^n is the number of days between September 11, 2016 and the nth date in the bitcoin data set
- d_n is the nth smallest element of $\{d_g^t\} \cup \{d_b^t\}$
- s_g^n is the nth residual from the gold ARIMA model
- s_b^n is the the nth residual from the bitcoin ARIMA model
- D_g is total number of days with recorded data for gold
- D_b is total number of days with recorded data for bitcoin
- p_g^n is the percentile of s_g^n in the set $\{s_g^1, \ldots, s_g^n\}$
- p_b^n is the percentile of s_b^n in the set $\{s_b^1, \ldots, s_b^n\}$
- θ is a constant to be optimized.

The autocorrelation and partial autocorrelation functions of the ARIMA model residuals for both gold and bitcoin suggest that the residuals are independent. Thus assuming the gold residuals are all sampled from the same distribution and the same is true of the bitcoin residuals, we can treat the collection of all residuals to date ($\{s_g^1, \ldots, s_g^{n-1}\}$ for gold and $\{s_b^1, \ldots, s_b^{n-1}\}$ for bitcoin) as random samples from a population of independent, identically distributed random variables.

We use the value of the empirical distribution function evaluated at s_g^n (p_g^n) to estimate the probability of getting a lower residual than s_g^n. By the central limit theorem, the empirical distribution converges pointwise to a normal distribution with $\sqrt{n}$ rate of convergence.[1] Thus, we introduce dampening factors $\sqrt{\frac{d_n - d_g^1}{D_g}}$ and $\sqrt{\frac{d_n - d_b^1}{D_b}}$ to account for the estimate uncertainty in p_g^n and p_b^n with small n.

The normal cumulative distribution function was considered, but neither the gold nor bitcoin residuals have high confidence of having been sampled from a normal distribution, as can be shown by the results of the Kolmogorov-Smirnov test[3] below:

```
        One-sample Kolmogorov-Smirnov test

data:  fit_gold$residuals
D = 0.4054, p-value < 2.2e-16
alternative hypothesis: two-sided
```

Figure 10: Kolmogorov-Smirnov test on gold residuals

```
        One-sample Kolmogorov-Smirnov test

data:  fit_gold$residuals
D = 0.4054, p-value < 2.2e-16
alternative hypothesis: two-sided
```

Figure 11: Kolmogorov-Smirnov test on bitcoin residuals

Thus the empirical distribution function was chosen for a better estimate of cumulative probabilities. We found that an approximate value of $\theta = 33.33$ yielded the highest final portfolio value when considering both the training set and the entire set of data. For the entire data set, this final value averaged to 71,562 USD, and for the training set it averaged to 15,723 USD.

5 Sensitivity Analysis

To test the robustness of our models, we will see how changing the values of α_{gold} and α_{bitcoin} affects the final values that each model generates at the end of the given time period.

We begin with slight changes to α_{gold} and α_{bitcoin}, and set them both to 0.03. In such a case, our best random model gives us an average final value of 8,515.3954 USD; our best proportional percentage model gives us a final value of 11,173.1433 USD; and our best EDF

model gives us a final value of 70,830.85 USD. As expected, the increased transaction costs cause our final values to decrease slightly across all models. However, it does not seem that these changes are drastic, and the ranking of model performance remains the same, with the EDF performing the best and the random model performing the worst. Considering only the EDF model, we obtain a final porfolio value of 55770.13 USD when $\alpha_{\text{gold}} = \alpha_{\text{bitcoin}} = 0.2$ and a final value of 58187.83 USD when $\alpha_{\text{gold}} = \alpha_{\text{bitcoin}} = 0.1$. We thus conclude that our algorithm is somewhat sensitive to transaction costs.

6 Strengths and Weaknesses

6.1 Strengths

We take into consideration how irregularities in prices might throw off our model analysis, and account for these irregularities (particularly the spike in bitcoin prices near the end) by examining the model performance over different segments of the time period. We also account for different transaction costs. Our best model, the EDF model, is consistently the best model, both across different values of α_{gold} and α_{bitcoin} as well as across different chunks of time from our given time period.

6.2 Weaknesses

Although we account for price irregularities when analyzing the performance of our models, we do so only briefly, and we could benefit from looking at how our model performs over more time intervals within the given period.

We should also test out more combinations of α_{gold} and α_{bitcoin}. This can help us find which models are most useful for traders who wish to more heavily invest in one asset or the other.

Our models rely on many assumptions, which do not always hold in real-world scenarios. Inflation is something which we do not account for but which has a great effect on the prices of the stock market. Similarly, we assume that if a trader has 0 USD or none of an asset, then they cannot trade it; however, in reality, traders are able to borrow money and take out loans. The core of our models also rely on traders buying and selling assets in a particular order, and only buying and selling each asset once per day. In reality, however, traders can conduct trades in any order they like, multiple times each day.

7 Conclusion

In conclusion, the EDF model consistently performs the best. Although our proportional percentage model also performs quite well at the end of the entire given time period–nearly as well as the EDF model–its performance is far worse than the random model for the first four-fifths of the time period, which indicates that the proportional percentage model's final value is largely bolstered by the bitcoin spike and that the model may be inaccurate in normal circumstances. The EDF model, however, performs much better than the random model for

both the entire time period as well as only 80% of it, which shows that it is more consistently profitable.

Appendices

Bibliography

[1] Empirical distribution function, 2022. URL https://en.wikipedia.org/wiki/Empirical_distribution_function.

[2] W. contributors. Ljung-box test, 2022. URL https://en.wikipedia.org/wiki/Ljung%E2%80%93Box_test.

[3] W. contributors. Kolmogorov-smirnov test, 2022. URL https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test.

To: Any trader whom it may concern

From: MCM Team # 2228316

Subject: Advice regarding optimal trading strategies

Date: Monday, February 21, 2022

Dearest Trader,

We have been hard at work analyzing the fluctuating prices of the gold and bitcoin markets in order to best assist you in your trading journey. We come to you now with significant findings, in hopes of guiding you in your quest to optimize your portfolio and maximize your profit.

We have developed three models that aim to predict the best trading options for each day, taking into account the market trends of the previous days. These three models consist of one which randomly assigns you percentages according to which you should buy and sell assets; one which takes into account the changes in today's prices and assigns you percentages based on these changes; and one which uses the empirical distribution function to guide its suggestions in how much of your assets you should trade. This latter model, which we have designated the Empirical Distribution Function (EDF) model, has proven to be the most useful. According to our data, if one had started with a portfolio of only 1,000 USD in September 11, 2016 and followed our EDF model for the next 5 years, then they could have increased the total value of their portfolio to 71,562 USD; a sizeable increase of 7,056.2%!

Upon conducting analysis upon this model, we find that it consistently performs well. We have tested this model over different periods of time, in order to account for sudden spikes in market prices, as well as with different transaction costs. No matter the situation, this model has displayed the best performance among the three, and has resulted in substantial profit. Of course, ,our research is not yet finished; in the future, we look forward to further testing the EDF model on an even wider variety of time frames and transaction costs, and we hope to expand it to markets beyond just bitcoin and gold.

Sincerely, MCM Team # 2228316