Detecting Hidden Regime in Cryptocurrency Price Trends with Hidden Markov Models

Project for the Course: STAT7102 Advanced Statistical Modelling

Group 25: WU Kwan-yu (UID:3035275501)

December 2020

1 Introduction

1.1 Background

Quantitative modelling and forecasting of asset price trends and returns have been a major area of interest for many quantitative researchers. With the technological advancement and improved computational power of machines, researchers have been focusing in algorithms using statistical learning methods in recent decades. There are various papers in examining the predictive power of historical asset prices in forecasting prices within the traditional financial asset spaces. The level of Efficient Market Hypothesis (EMH) that the particular asset class exhibits has also been a major subject in question.

On the other hand, cryptocurrency is a new form of "money" invented in recent years with a unique decentralized property. Viewing cryptocurrency as an asset, it is questionable to many whether its market is efficient with its own dynamic, and whether cryptocurrency price trends exhibits macroeconomic scenario states (ie. "hidden regimes") as in traditional assets such as stock and bonds, with clear market directions such as growth state, recession state, bearish state and state. Previous papers on the topic of cryptocurrency have found out a certain degree of market inefficiency in weak form. In this paper, the efficiency of cryptocurrency market is the major question, and Hidden Markov Model (HMM) is used to examine the possibility of generating outperforming strategy.

1.2 Project Objective

This project aims to carry out asset return modelling on several major cryptocurrencies in the market. We firstly investigate the data collected from Section 2.1 with a brief analysis of descriptive statistics in Section 2.3, and the statistical methodology of HMM is introduced in Section 3. The possible hidden regime and regime switching in these coin markets using multivariate hidden Markov model (MHMM) techniques is examined in Section 3 with result interpretation in Section 3.2. Finally, a simple long-short trading strategy is suggested in Section 4, aiming at displaying the potential of the model in detecting the underlying regime in the market.

Coin	Mean	Stdev	Skewness	Kurtosis
BTC	0.1319%	0.0417	-1.0813	18.7622
ETH	0.0607%	0.0511	-1.2230	16.3172
XRP	0.1032%	0.0606	1.7060	22.7405
LTC	0.0366%	0.0552	0.4084	12.5271
LINK	0.3652%	0.0781	0.1872	9.9196

Table 1: Return distributions

2 Data

2.1 Data Collection

The historical daily closing price dataset for this project is obtained from CoinMarketCap (https://coinmarketcap.com). Five large coins in terms of market capitalization are involved, namely Bitcoin (BTC), Ethereum (ETH), XRP (XRP), Litecoin (LTC) and Chainlink (LINK). The time period under investigation is October 2017 to December 2020. Earlier time points of data are not taken into account due to the data availability of all coins, and the immaturity of cryptocurrency market in earlier periods.

2.2 Data Preprocessing

To facilitate statistical modelling in upcoming sections, some features from the original data are generated. The change in price and logarithmic return is calculated by 1

$$\text{Price Change}_t = \frac{\text{Price}_t}{\text{Price}_{t-1}} \tag{1}$$

$$\text{Log-Return}_t = \ln(\text{Price Change}_t) \tag{2}$$

In order to investigate price trends in various market sensitivity means, three separate dataset with various sampling period is generated from the obtained daily closing price. The three dataset corresponds to 1-day closing price, 3-day closing price aggregation, and 5-day closing price aggregation. It is expected that a greater aggregation provides macroeconomic insight in slower-moving horizon because of lessened price fluctuation and market noises.

2.3 Descriptive Statistics

This section aims to provide a statistical outlook of the obtained data.

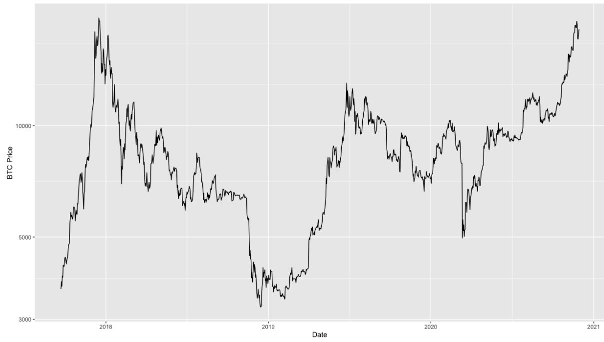

Figure 1: BTC Historical Log-Price

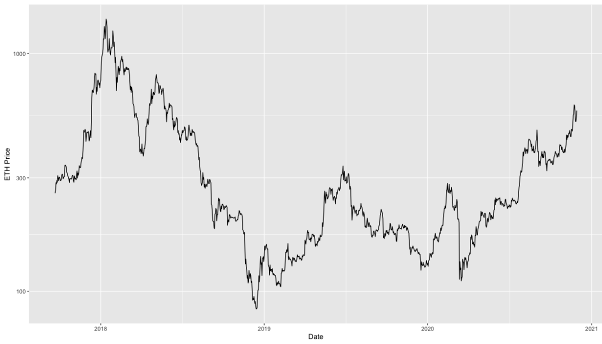

Figure 2: ETH Historical Log-Price

Figure 1, 2, 3 and Table 1 suggested the highly volatile nature of the cryptocurrency market, which is consistent with the well-known characteristics of cryptocurrency market. Furthermore, the slightly positive mean returns are with fat tails of the five coins as reflected from kurtosis statistics implies the possibility of extreme earnings or losses from investing in these coins.

3 Methods

Hidden Markov Model is an unsupervised leaning method which consists of a pre-determined of states, where states are linked by transition probabilities and each states have it own set of distribution parameters. In this project, Hidden Markov Model (HMM) is adopted to model cryptocurrency returns as it is hypothesized that states of economic scenarios exist, which drift the price trend. For a single time series price data, a univariate HMM is appropriate. To factor out unique characteristics of a certain coin and filter out market characteristics, a multivariate HMM is required.

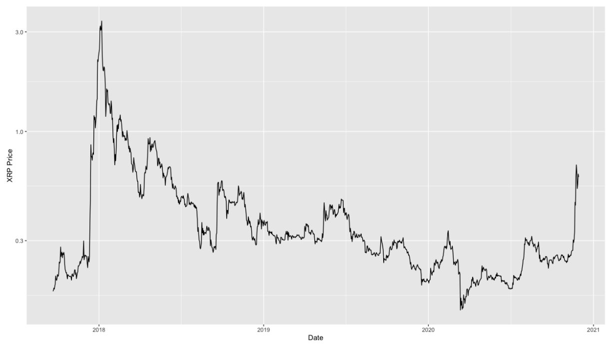

Figure 3: XRP Historical Log-Price

3.1 Hidden Markov Model

Given a time-correlated series of observation $x_1, x_2, ...x_t$, a Hidden Markov Model consists of the following characteristics:

- An underlying hidden state path with k states in $s_1, s_2, ..., s_t$.
- A state transition matrix A_{ij} where a_{ij} is the probability of state transition from i to j.
- Posterior probabilities $\pi_t = P(x_t|s_t)$ reflecting the probability of x_t being at state s_t in time t.

The objective of the model is to estimate the set of parameters for the underlying distributions and transition probability, by obtaining a marginal maximized likelihood for the observed sequence. The computational detail of HMM is beyond the scope of this project and hence skipped.

The model can be further extended to incorporate multivariate response observations $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_t$, with hidden states $s_1, s_2, ..., s_t$, and different distribution assumption. In this project, n-dimensional response vector is chosen for investigation where n is the number of coins to incorporate, representing the overall cryptocurrency market outlook. Also, the responses are assumption to follow Gaussian distributions.

The biggest question of this project is to find a choice of number of states for an adequate model, as well as investing on the effects of time period aggregation, given a series of observation of log-return data, so that macroeconomic factors can be well-understood by interpreting the model outcome. In the following Section 3.2, various choices of parameters will be set for comparison.

	State 1	State 2	State 3	State 4
State 1	0.346	0.187	0.075	0.392
State 2	0.287	0.252	0.091	0.369
State 3	0.112	0.262	0.466	0.160
State 4	0.357	0.105	0.014	0.524

Table 2: Transition matrix for 4-state 3-coins model

	Coin 1 (BTC)		Coin 2 (ETH)		Coin 3 (XRP)	
	Mean (Intercept)	Stdev	Mean (Intercept)	Stdev	Mean (Intercept)	Stdev
State 1	0.057	0.046	0.065	0.043	0.046	0.041
State 2	-0.084	0.037	-0.121	0.049	-0.107	0.053
State 3	0.029	0.145	0.080	0.154	0.142	0.262
State 4	-0.006	0.025	-0.013	0.029	-0.013	0.029

Table 3: Parameter estimates for 4-state 3-coins model

3.2 Results

3.2.1 4-State MHMM

The most adequate model is firstly presented in this section, followed by other comparable results. It is found that a 3-day aggregated, 3-coin-4-state model is the most adequate MHMM in detecting hidden states of cryptocurrency returns.

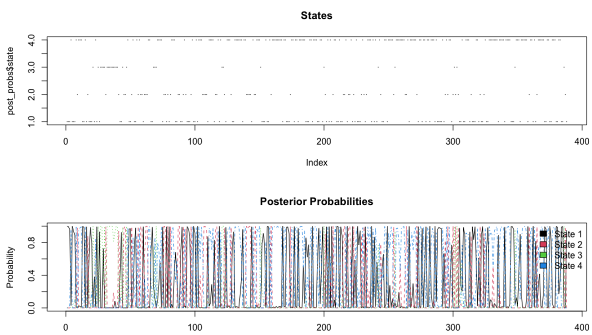

Figure 4: State path and posterior probabilities for 4-state 3-coins model

The transition matrix is as shown in Table 2. The transition matrix reflects a notable likelihood of transitions between each states, and the historical states and posterior probabilities during the time period is given by Fig 4.

The model reports a maximum log-likelihood of 1702.678 and Akaike Information Criterion (AIC) of -3327.356. The estimated parameters for distribution of different states are given by Table 3.

	Coin 1 (BTC)		Coin 2 (ETH)	
	Mean (Intercept)	Stdev	Mean (Intercept)	Stdev
State 1	0.000	0.064	-0.001	0.078
State 2	0.002	0.015	-0.001	0.021

Table 4: Parameter estimates for 2-state 2-coins model

It can be seen from the parameter estimates that the model is successful in classifying trends into different macroeconomic scenarios resulting in different return distributions: (1) a bullish market represented by State 1 with positive mean return and average volatility, (2) a bearish market represented by State 2 with negative mean return and average volatility, (3) a thriving market represented by State 3 with highly positive mean return yet with high volatility and (4) a steady market represented by State 4 with near-zero mean return and low volatility. The three coins produces similar return distribution under the same state, despite their size differences in market capitalization. This shows a consistent macroeconomic effect on different coins.

Moreover, the choice of $k = 4$ as the pre-determined number of states produced a consistent result to previous suggestions from the academia (Guidolin & Timmermann, 2007) that a 4-state model outperforms other models with different number of states ($k = 1, 2, ..., 6$) in explaining macroeconomic trends, despite the asset-in-question is different. Such behavior of asset return trends further supports the statement of cryptocurrency as an alternative to traditional financial assets.

3.2.2 2-State MHMM

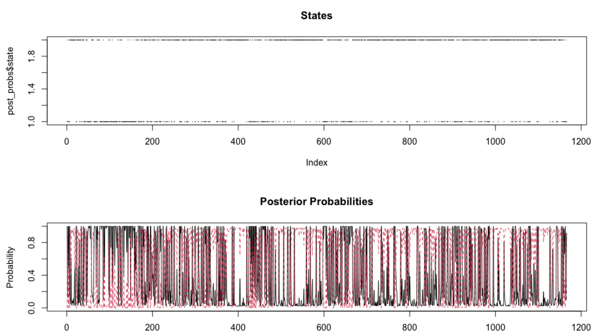

Figure 5: State path and posterior probabilities for 2-state 2-coins model

An alternative choice of parameters produces a 2-state 2-coin model. The model reports a maximum log-likelihood of 4403 and AIC of -8783.988, with a high level of certainly in posterior probability as shown in 5, suggesting an adequate fit of the model. However, the choice in number of state $k = 2$ does not classify returns with macroeconomic varieties.

It can be seen from the parameter estimates shown in 4 that (1) State 1 represents a fanatic market with near-zero mean return and high volatility, and (2) State 2 represents a calm market with near-zero mean return and low volatility. The 2-state model generates market insight, only in terms of market volatility, instead of underlying regimes and hence, a less ideal model fit.

4 Model Application

In Section 3.2, an adequately fitted MHMM is introduced. In this Section, a simple equal-weighted long-short trading strategy based on the fitted MHMM will be introduced in order to examine the forecasting power of the model. It can be shown from the backtesting results that the trading strategy outperforms the benchmark.

4.1 Simple Trading Strategy

A set of parameters is chosen for the trading strategy setup:

- Aggregation period: 3-day aggregation method is used as it is able to effectively reducing market noises from 1-day model.
- Coins: three coins (BTC, ETH, XRP) are chosen as they have large market capitalization and is relatively mature in the market.
- Number of states: 4 states are chosen for the MHMM to simulate the economic regimes.
- Backtesting period: from October 2017 to December 2020.
- Risk-free rate: assumed constant at 0.12%.

The parameters choices are identical to the 4-state model introduced in Section 3. The trading strategy is as follows:

- The portfolio consists of 3 coins, equally weighted for simplicity in demonstration.
- Switch to short position at time $(t+1)$ when the underlying state at time t is 2.
- Switch to long position at time $(t+1)$ when the underlying state at time t is not 2.

The backtesting result is then compared to the benchmark performance, which is a buy-and-hold strategy for a portfolio consisting the same coins, also equally weighted.

4.2 Backtesting Result

Table 5 and Fig 6 shows the backtesting result and its performance comparing to the benchmark. The simple trading strategy outperforms the benchmark in terms of log-return and Sharpe ratio, and have a similar volatility as the benchmark. The strategy is able to

	Log-return	Volatility	Sharpe Ratio
Strategy	1.9488	0.0799	24.3526
Benchmark	1.5470	0.0800	19.3168

Table 5: Performance metrics of backtest results

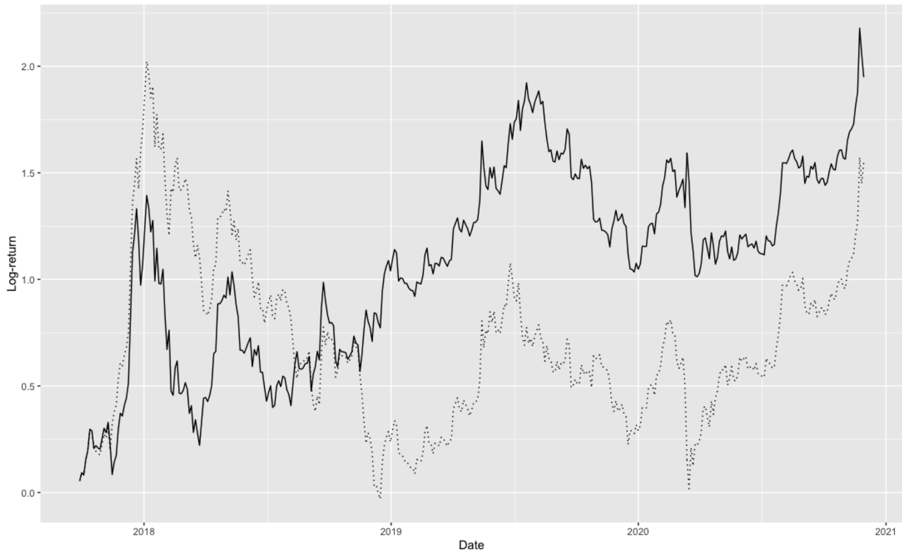

Figure 6: Log-return backtesting results (Solid line: Strategy, Dotted line: Benchmark)

detect a bearish market and hence switching its winning position. It is undoubtedly that the model exhibits a certain degree of predictive power in forecasting future cryptocurrency market trends.

5 Conclusion

In this project, a Multivariate Hidden Markov Model is introduced in order to detect if a hidden macroeconomic regime exists that influences the cryptocurrency market trends. In Section 3 the details of the statistical model is provided and the model is specified and evaluated in Section 3.2. The model is applied as a simple trading strategy and a backtesting is carried out in Section 4 which proves the significance of the fitted model and its usefulness in forecasting future cryptocurrency market trends. Yet, the investigation on model significance shall be continued as the availability of time points is short as compared to other financial markets such as bonds since cryptocurrency is currently still a newly emerged market. In addition, there are possibly not enough macroeconomic scenarios appear during the period of investigation.

References

Guidolin, M., & Timmermann, A. (2007). Asset allocation under multivariate regime switching. *Journal of Economic Dynamics and Control*, *31*(11), 3503–3544.

Gupta, A., & Dhingra, B. (2012). Stock market prediction using hidden markov models. *2012 Students Conference on Engineering and Systems*, 1–4.

Kim, E.-c., Jeong, H.-w., & Lee, N.-y. (2019). Global asset allocation strategy using a hidden markov model. *Journal of Risk and Financial Management*, *12*(4), 168.

Koki, C., Leonardos, S., & Piliouras, G. (2019). A peek into the unobservable: Hidden states and bayesian inference for the bitcoin and ether price series. *arXiv preprint arXiv:1909.10957*.

Nelder, J. A., & Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, *135*(3), 370–384.

Nguyen, N. (2018). Hidden markov model for stock trading. *International Journal of Financial Studies*, *6*(2), 36.

Noda, A. (2020). On the evolution of cryptocurrency market efficiency. *Applied Economics Letters*, 1–7.

Visser, I., Speekenbrink, M. et al. (2010). Depmixs4: An r package for hidden markov models. *Journal of Statistical Software*, *36*(7), 1–21.

Appendices

A Related R Codes

```
library(dplyr)
library(depmixS4)
library(data.table)
library(ggplot2)
library(moments)

# Feature Engingeering -----------------------------------------------------

coins_data <- read.csv('./data/coins_data.csv')

# calculate simple pct change and log returns
calculate_return <- function(coins){
  ndf <- nrow(coins)
  # btc
  c <- 2
  coins$btc_ret <- c(NA, (coins[2:ndf, c])/coins[1:(ndf-1), c])
  coins$btc_logret <- log(coins$btc_ret)

  # eth
  c <- 3
  coins$eth_ret <- c(NA, (coins[2:ndf, c])/coins[1:(ndf-1), c])
  coins$eth_logret <- log(coins$eth_ret)

  # xrp
  c <- 4
  coins$xrp_ret <- c(NA, (coins[2:ndf, c])/coins[1:(ndf-1), c])
  coins$xrp_logret <- log(coins$xrp_ret)

  # ltc
  c <- 5
  coins$ltc_ret <- c(NA, (coins[2:ndf, c])/coins[1:(ndf-1), c])
  coins$ltc_logret <- log(coins$ltc_ret)

  # link
  c <- 6
  coins$link_ret <- c(NA, (coins[2:ndf, c])/coins[1:(ndf-1), c])
  coins$link_logret <- log(coins$link_ret)

  # drop first row with NA value
  coins <- na.omit(coins)
  return(coins)
}

# aggregate price data on 1-day 3-day and 7-days (weekly)
n <- nrow(coins_data)
# 1d
```

```
agg <- 1
coins1d <- coins_data
coins1d$group <- rep(1:(n%/%agg), each=agg)
coins1d <- calculate_return(coins1d)

# 3d
agg <- 3
coins3d <- coins_data
coins3d$group <- rep(1:(n%/%agg), each=agg)
coins3d <- coins3d %>% group_by(group) %>% slice_tail()
coins3d <- calculate_return(as.data.frame(coins3d))

# 7d
agg <- 7
coins7d <- coins_data
coins7d$group <- c(rep(1:(n%/%agg), each=agg), rep(n%/%agg+1, each=n%%agg))
coins7d <- coins3d %>% group_by(group) %>% slice_tail()
coins7d <- calculate_return(as.data.frame(coins7d))


# Descriptive Statistics -----------------------------------------------------

ggplot(coins1d, aes(x=as.Date(date))) +
  geom_line(aes(y = btc_price)) +
  labs(x="Date", y="BTC Log-Price") +
  scale_y_log10()

ggplot(coins1d, aes(x=as.Date(date))) +
  geom_line(aes(y = eth_price)) +
  labs(x="Date", y="ETH Log-Price") +
  scale_y_log10()

ggplot(coins1d, aes(x=as.Date(date))) +
  geom_line(aes(y = xrp_price)) +
  labs(x="Date", y="XRP Log-Price") +
  scale_y_log10()

stats <- function(col){
  col <- na.omit(col)
  stat <- data.frame(mean(col), sd(col), skewness(col), kurtosis(col))
  names(stat) <- c('Mean','SD','Skewness','Kurtosis')
  return(stat)
}

des <- rbind(stats(coins1d$btc_logret),
             stats(coins1d$eth_logret),
             stats(coins1d$xrp_logret),
             stats(coins1d$ltc_logret),
             stats(coins1d$link_logret))
```

```
row.names(des) <- c('BTC','ETH','XRP','LTC','LINK')
des

# Multivariate HMM 3D aggregation---------------------------------------------
set.seed(2020)

# hmm3d3c4s: 3 Days, Top 3 Coins, 4 states model
hmm<- depmix(list(btc_logret~1,
                  eth_logret~1,
                  xrp_logret~1),
              nstates=4,
              family=list(gaussian(),
                          gaussian(),
                          gaussian()),
              data=coins3d)

hmmfit <- fit(hmm, verbose=FALSE)
# em.control=em.control(maxit=500, tol=1e-8,random.start=TRUE)
post_probs <- posterior(hmmfit)
summary(hmmfit)
hmmfit

# plotting regime switch
par(mfrow = c(2, 1))
plot(post_probs$state,
     main='States',
     pch=16,
     cex=0.2)
matplot(post_probs[,-1],
        type='l',
        main='Posterior Probabilities',
        ylab='Probability')
legend(x='topright',
       c('State 1','State 2','State 3','State 4'),
       fill=1:4,
       bty='n')

coins3d$state <- post_probs$state


# Multivariate HMM 1D aggregation 1 coin--------------------------------------
set.seed(2020)

# hmm3d3c4s: 3 Days, Top 3 Coins, 4 states model
hmm<- depmix(list(btc_logret~1,
                 eth_logret~1),
             nstates=2,
             family=list(gaussian(),
                         gaussian()),
             data=coins1d)
```

```
hmmfit <- fit(hmm, verbose=FALSE)
# em.control=em.control(maxit=500, tol=1e-8,random.start=TRUE)
post_probs <- posterior(hmmfit)
summary(hmmfit)
hmmfit

# plotting regime switch
par(mfrow = c(2, 1))
plot(post_probs$state,
     main='States',
     pch=16,
     cex=0.2)
matplot(post_probs[,-1],
        type='l',
        main='Posterior␣Probabilities',
        ylab='Probability')
legend(x='topright',
       c('State␣1','State␣2','State␣3','State␣4'),
       fill=1:4,
       bty='n')

coins3d$state <- post_probs$state


# BACKTESTING ----------------------------------------------------------------

N <- 3

btc_decision <- function(s){
  decision <- c(1,-1,1,1)
  result <- (decision[s])
  return(result)
}

eth_decision <- function(s){
  decision <- c(1,-1,1,1)
  result <- (decision[s])
  return(result)
}

xrp_decision <- function(s){
  decision <- c(1,-1,1,1)
  result <- (decision[s])
  return(result)
}


portfolio_calculation <- function(coins3d){
  coins3d$equalval_ret <- (coins3d$btc_ret+coins3d$eth_ret+coins3d$xrp_ret)/N
```

```
  coins3d$equalval_logret <- log(coins3d$equalval_ret)
  coins3d$btc_decision <- btc_decision(coins3d$state)/N
  coins3d$eth_decision <- eth_decision(coins3d$state)/N
  coins3d$xrp_decision <- xrp_decision(coins3d$state)/N

  coins3d$btc_decision_lag1 <- shift(coins3d$btc_decision, n=1)
  coins3d$eth_decision_lag1 <- shift(coins3d$eth_decision, n=1)
  coins3d$xrp_decision_lag1 <- shift(coins3d$xrp_decision, n=1)

  coins3d$btc_payoff = (coins3d$btc_ret - 1) * (coins3d$btc_decision_lag1)
  coins3d$eth_payoff = (coins3d$eth_ret - 1) * (coins3d$eth_decision_lag1)
  coins3d$xrp_payoff = (coins3d$xrp_ret - 1) * (coins3d$xrp_decision_lag1)

  coins3d$port_ret = coins3d$btc_payoff+coins3d$eth_payoff+coins3d$xrp_payoff+1
  coins3d$port_logret = log(coins3d$port_ret)

  coins3d <- na.omit(coins3d)
  coins3d$date <- as.Date(coins3d$date)

  return(coins3d)
}

coins3d <- portfolio_calculation(coins3d)

ggplot(coins3d, aes(x=date)) +
  geom_line(aes(y = cumsum(port_logret))) +
  geom_line(aes(y = cumsum(equalval_logret)), linetype="dotted") +
  labs(x="Date", y="Log-return") +
  scale_fill_manual(labels = c('Strategy','Equal-weighted␣Benchmark'))


port_logret <- sum(coins3d$port_logret)
equalval_logret <- sum(coins3d$equalval_logret)

port_vol <- sd(coins3d$port_logret)
equalval_vol <- sd(coins3d$equalval_logret)

Rf = 0.12/100
port_sharpe <- (port_logret - Rf)/port_vol
equalval_sharpe <- (equalval_logret - Rf)/equalval_vol

port_metrics <- cbind(port_logret, port_vol, port_sharpe)
equalval_metrics <- cbind(equalval_logret, equalval_vol, equalval_sharpe)
metrics <- data.frame(rbind(port_metrics, equalval_metrics))
names(metrics) <- c('Log-return', 'Volatility', 'Sharpe')
row.names(metrics) <- c('Portfolio','Equal-weighted␣Benchmark')
metrics
```