INVICTUS ALPHA

Invictus Alpha Graduate Program Quantitative Assessment

Candidate:

Kayla Siobhan Hopley

An assessment submitted in fulfilment of the requirements for the quants internship

---

Question 1: Theory

Time taken to complete: 3 hours

1.1.1

Q:

How are the costs for a transaction on the Ethereum network determined? What are the key factors behind these costs? Include time-varying factors as well as constant factors in your answer

A:

Ethereum runs on something called 'gas', it is a unit used to measure the computational power required in order to execute certain operations. These operations are processed by miners and completed on the Ethereum blockchain. If we consider the everyday activity of driving a car for example, the desired operation is to drive it, the fuel used to fill your car is the gas and the gas station is the miner. Much like fuel, gas isn't a currency on its own and is therefore given a value (gas is expressed in Ether, the Ethereum network's underlying currency).

To get a rough estimate one can view the table found in the Ethereum yellow paper. Every transaction requires roughly $21,000$ gas. As Ethereum is a proof-of-work peer-to-peer system it is heavily dependent on the hashrate of their miners. The more miners, the more hashrate, the more secure and faster the system is. Miners earn money by mining blocks and getting block awards. They are the dictators of their mined blocks and are responsible for putting transactions inside their blocks.

To price gas, which fluctuates depending on network demand, the Ethereum network uses a "first price auction" system. It works in that each sender submits a bid (also known as a gas limit) for how much they are willing to pay for the miner to include their transaction in the next block. It is important that the sender chooses a gas limit that is within the miner's threshold, otherwise the miner may choose to not execute that transaction. Additionally, senders can pay higher fees to prioritise their transaction's inclusion in the next block while receiving refunds should their transaction be rejected by miners.

See: [1], [2]

1.1.2

Q:

When trading, under what circumstances would you need to take these fees into account?

A:

Gas fees are an important component of the Ethereum network, as they are the fees required by miners to execute transactions. As previously mentioned, if the gas limit set by the user does not meet the miners' threshold, the transaction will either be delayed or rejected and not included in the next block, making it nearly impossible for $ERC20$ projects to run even microtransaction payments on the network.

You would need to take these fees into account when the Ethereum network is plagued with network congestion and by extension rising fees (this was a problem when the network was inundated with NFT's). Congestion arises from an increased number of transactions like depositing assets on lending platforms, trading on decentralised exchanges, or sending tokens.

When Uniswap announced an airdrop of its new UNI token, many investors rushed to trade UNI on their platform. This resulted in a large and sudden increase in gas fees as users competed for block space. Circumstances like these are certainly ones to be aware of when trading on the Ethereum network.

See: [3], [4], [5], [6], [7]

1.1.3

Q:

What is an Automatic Market Maker? How do these differ from regular exchanges?

A:

Traditional (centralised) exchanges rely on buyers and sellers to provide liquidity, in contrast to decentralised exchanges in decentralised finance, which trade 24/7, relying on mechanisms known as automated market makers (AMM). AMM's are smart contracts (algorithms) that provide liquidity to decentralised exchanges via automated trading that instantly finds liquidity for those wanting to buy or sell a digital asset. The algorithm, which makes use of liquidity pools (cryptocurrencies that are lent to the protocol by liquidity providers) removes the need for a "centralised" type of exchange used by traditional markets. AMMs have therefore replaced the traditional limit order-book (used in traditional markets) with a mechanism whereby digital assets are automatically swapped against the latest price of the asset in the liquidity pool.

See: [8], [9], [10], [11], [12], [13], [14], [15]

1.1.4

Q:

What is a flash loan? How is the lender guaranteed the loan is returned?

A:

A flash loan is a fast growing financial product that is becoming popular across a number of decentralised finance protocols on the Ethereum network. A relatively new instrument, flash loans allow users to instantly borrow collateral-free capital loans as long as the borrowed capital, plus a fee, is returned before the end of the blockchain transaction. The entire process takes seconds;

1. Transaction gets submitted to the Ethereum network
2. The funds are temporarily borrowed
3. The borrower checks if the trade is profitable, and if not rejects the translation resulting in funds being returned to the lender

Importantly, the funds are always returned to the lender in either case (profitable or not).

As the entire transaction is recorded in one blockchain transaction, the lender's funds are guaranteed to be returned as according to the digital ledger, the funds never left the lender in the first place.

See: [16], [17], [18], [19], [20], [21], [22]

1.2

Q:

What is a stable coin? What is the role that they play in the crypto ecosystem? Explain the different means used to construct stablecoins.

A:

A stablecoin (e.g. Tether ($USDT$)) is a type of digital currency that attempts to peg it's market value to another asset. They are pegged to fiat currencies (e.g. the dollar) or commodities such as gold e.g. Tether Gold ($XAUt$). While there exists three types of collateralisation used by stablecoins to ensure price stability, the most commonly used form is where the organisation that governs the stablecoin sets up a reserve where it stores the asset the stablecoin is pegged to e.g. \$10m in bank deposits to back up \$10m in stablecoin prints. Other forms include stablecoins that are collateralised by other crypto assets rather than by fiat currencies. MakerDAO utilises this method through collateralised debt positions, where an individual deposits and locks up his/her crypto assets as collateral with DAI (the stablecoin of MakerDAO) is then minted. The third and least popular form of stablecoin is an algorithmic stablecoin, where because no collateralisation is used, coins either created or burned to keep the asset's value in line with its targeted price.

Stablecoins have seen a surge in popularity as a result of its promised lower volatility which makes them useful for trading between fiat currencies and other cryptocurrencies. Much like how investors not involved in crypto will allocate a portion of their portfolio to cash, money markets funds, or bonds during times of heightened volatility, cryptocurrency investors move to stablecoins to diversify their portfolio. While moving to stablecoins allows investors to stay in the crypto market and protect the value of their investment, it also allows them to move with speed between trades without having to deposit from fiat currency back into crypto. Not to mention that investors can earn interest on their deposits, and often at much higher rates than those offered by banks.

The advent of stablecoins and the provision of a "risk-off" instrument has been one of the contributing factors to the growth of the industry.

See: [23], [24], [25], [26], [27], [28], [29], [30], [31]

1.3

Q:

What is a perpetual future? Explain one arbitrage opportunity that a perpetual future may uniquely provide.

A:

A traditional futures contract is an agreement between two parties who commit to purchasing or selling the underlying assets at a predetermined price on a future date. A perpetual future on the other hand is a unique form of future that has no expiration / settlement date. As it has no expiry date, traders are able to hold their position open for as long as they wish. Unlike traditional futures, traders generally trade perpetuals at prices equal to spot. However, often the price of the perpetual, which is based on the average price of the asset, is not equal to the spot price of the asset. It is as a result of this difference in price that an arbitrage opportunity arises.

Like with traditional futures, prices of perpetuals must also converge to the spot price in the future. As perpetuals have no expiration date, a funding mechanism is used to ensure the price of the perpetual converges to the underlying spot price. It works that if the perpetual price is greater than the spot price, funding rate is positive and longs pay the shorts. Conversely if the perpetual price is less than spot, the funding rate is negative and shorts pay the longs. The idea is that due to leverage used by traders, liquidations will occur pushing the perpetual price closer to the spot price.

As mentioned above, the arbitrage opportunity exists due to price convergence as over time prices of futures and spot will converge to the same price.

Note: Example of arbitrage trading opportunity below assumes zero fees.

Year 1:

- Eth perpetual price: $\$2,000$
- Eth spot price: $\$1,900$

To execute this trade the trader would short the perpetual and purchase the same amount of the asset on the spot exchange. We assume order size to be 1 Eth.

Year 2 (assuming prices converge):

- Eth perpetual price: $\$2,200$
- Eth spot price: $\$2,200$

The trader closes the position and realises the following profit:

$-\$200$ (loss on perpetual) $+$ $\$300$ (profit on spot) $=$ $\$100$ profit

See: [32], [33], [34], [35], [36], [37]

Question 2: Problem Solving

Time taken to complete: 4.5 hours (Please note there is an extra zero in the total_capital)

2.2

Problem:

You need to rebalance a crypto index fund that contains an asset cap. This means that the fund's constituents are rebalanced according to their market capitalisations - until they reach their cap. The remaining funds that would have been allocated above the cap are then redistributed. Make sure no other asset exceeds its cap. This should work for any number of assets and any cap. First calculate the percentage allocations and then capital allocations and amounts. A good sanity check is to look at the outputs with an unlimited cap (1 or 100%)

Solution:

- Start off by inputting the assets in the crypto index fund, there is no limit to how many can be put in. 'Ticker' (currency symbol) in the first column, 'MCAP' (market cap) in the next and finally 'Price' in the last. Data saved as a table. (Sort by highest MCAP)

- Calculate the ratio of the market cap for every currency by summing all market caps (total_MCAP) and then dividing each market cap by the sum. Save this as a new column called 'MCAP_ratio'
- Ask for the user to input their desired asset cap (this should be between 0 and 1) and the total capital (this is the total amount in USD the user wants to invest)
- The maximum allowed to be invested into any asset is given by the asset cap multiplied by the total capital → max_invest = asset_cap ∗ total_capital
- If the asset cap < (1 / number of assets) then the amount invested into each asset should simply be total_capital ∗ (1 / number of assets). If the asset cap > (1 / number of assets) then we can move onto the next step
- If the MCAP_ratio is >= greater than or equal to the asset cap the value of USD invested into that asset is max_invest
- Next we'll need to subtract the market cap of the asset we just invested in from the total_MCAP to give us a new_total_MCAP. We also need to subtract the (max_invest) from the total_capital to give the amount of remaining USD we have to invest. We will then recompute the market cap ratios of the remaining assets with the new_total_MCAP
- The max asset cap can't be exceeded now so we simply need to multiply the rest of the market cap ratios by our remaining USD amount to get the value we will invest into every asset.
- Once we have the amount invested in USD into every asset we can calculate the amount of each asset we received by dividing the amount_invested in USD by the price

In the first example we have:

- total_MCAP = $35,000$
- MCAP_ratio = 0.57 BTC, 0.29 ETH, 0.14 LTC
- asset_cap = 0.5
- total_capital = $1,000$ (Note: the question says $10,000$ but this makes no sense with the output)
- max_invest = $0.5 * 1,000 = 500$
- As the asset_cap is > (1 / number of assets) i.e. 0.5 > (1 / 3) we can move onto the next step
- BTC MCAP 0.57 > asset_cap = 0.5 so we invest the max allowed = 500
- new_total_MCAP = total_MCAP - BTC MCAP = $15,000$
- rem_amount = total_capital - max_invest = 500
- Ratios of the MCAP's in relation to new_total_MCAP are 0.66 ETH and 0.33 LTC
- Multiply rem_amount by the new ratios to get amount invested 333.33 ETH and 166.67 LTC
- Divide the amount_invested by the price to get the amount of each asset 10.000000 BTC, 13.333333 ETH, 16.666667 LTC

In the second example we have:

- The asset_cap is < (1 / number of assets) i.e. 0.1 < (1 / 3) we can simply take total_capital and $*$ (1 / number of assets) so $1,000$ * (1/3) = 333.33 invested into BTC, ETH and LTC
- Divide the amount_invested by the price to get the amount of each asset 6.666667 BTC, 13.333333 ETH, 33.333333 LTC (all percent values should be equal)

Please see rebalancingCIF.py (click me) for the coded solution.

Question 3: Analysis

Time taken to complete: 11 hours

This question requires you to use the FTX exchange API. Pull 1 year's worth of historical data for spot BTC/USD and the BTC/USD perpetual future in 5 minute OHLCV format. Do the same for ETH/USD.

To complete this I used ccxt which can be found here: [41]

In my code FTXapi.py (click me) I created two functions. One pulls the spot BTC/USD and then the BTC-PERP in 5 minute OHLCV format and saves both to CSV. The second does the same but for ETH. The first OHLCV CSV for both is labelled as raw data as it needs to be converted to date time.

See also: [38], [39], [40], [41], [42], [43], [44], [45], [46]

3.1

Problem:

Calculate the sharpe and sortino ratios of both of the assets. What can you conclude from your results?

Solution:

The sharpe ratio is defined as

$$Sharpe = \frac{\overline{R}_p - R_f}{\sigma_p} \tag{1}$$

Where $\overline{R}_p$ is the mean portfolio returns, R_f is the risk free rate, σ_p is the risk, standard deviation, of the portfolio. This is for assets such as stocks. In crypto we can simply

- Plot time series data for a set of annual returns
- Calculate mean average of the returns
- Measure the squared average distance from the mean
- Take the square root giving us the standard deviation

$$Sharpe = \frac{AverageReturn}{StandardDeviation} \tag{2}$$

This ratio penalises upside and downside volatility equally. BTC has a currently sharpe ration of roughly 3.5 and ETH just below that. Previously ETH's was higher but using the sharpe ratio alone isn't a good idea. We also need to take into account the drawdown. The risk of investing in ETH is higher despite its current sharpe ratio being lower.

The Sortino ratio is defined as:

$$Sortino = \frac{\overline{R} - \overline{R}_f}{LPSD} \tag{3}$$

Where $LPSD$ is the Lower Partial Standard Deviation. Unlike the sharpe ratio the sortino ratio only takes downward deviation into account

- Plot time series data for a set of annual returns
- Calculate mean average of the returns

- Calculate the standard deviation only when the return is negative or below the baseline for minimum expected return

$$Sortino = \frac{AverageReturn}{DownsideDeviation} \tag{4}$$

This means that any positive return will not negatively effect the rating. BTC and ETH current sortino ratio's are below 1 $\rightarrow$ 0.05 and 0.06 respectively.

At the moment both ratios are very similar so I do not think these solely are good indicators of what to invest in.

See: [47], [48], [49]

3.2

Problem:

Calculate the information ratio of BTC with respect to ETH as a benchmark and interpret your result.

Solution:

The Information ratio seeks to evaluate risk-adjusted return in relation to a baseline investment. It is a performance measure when being compared to a benchmark.

$$InformationRatio = \frac{ExcessReturn - BenchmarkReturn}{TrackingError} \tag{5}$$

As the market is in a bit of a slump both IR's will low or negative meaning they are failing to yield a return.

3.3

Problem:

Calculate the tracking error (over full series) and the rolling 7-day tracking error of each spot vs each perpetual. Interpret your results.

Solution:

The tracking error can be described as the relative risk

$$TrackingError = \sigma(PortfolioReturns - BenchmarkReturns) \tag{6}$$

Tracking error of full series will be higher than the tracking error over 7-day (underperformed). The tracking error helps measure and compare performance of a portfolio and allows you to gauge the consistency of excessive returns.

See: [50]

Question 4: Engineering and Market Operations

This question requires you to create an account on Coinbase Pro and Google Cloud Platform.

4.1

Problem:

Create a program to interact with the Coinbase Pro sandbox exchange and place orders. You may implement a simple strategy or place orders randomly. Record your orders, current positions and balances in a database. Make this information available via a live API hosted on GCP. Create an endpoint for each.

Solution:

Struggled to verify my Coinbase account with a German driver's / Irish passport. See: [51] for the code I was looking at using.

Bibliography

[1] Luno. 'Understanding Ethereum fees: How gas works'

[2] GlassNode. 'Ethereum: Total Transaction Fees'

[3] Coindesk. 'High gas fees prevent Ethereum from being Ethereum'

[4] Cointelegraph. 'Ethereum fees are skyrocketing but traders have alternatives'

[5] Tangany. 'How to avoid high gas fees on transactions'

[6] Fxstreet. 'How tis Ethereum tackling it's gas fee problem'

[7] Fool. 'The 4 biggest mistakes you can make when buying Ethereum'

[8] Gemini. 'What are automated market makers?'

[9] Binance. 'What is automated market maker?'

[10] CMC. 'AMM'

[11] CMC. 'What are automated market makers?'

[12] Medium. 'AMM breakdown'

[13] Blockchain. 'Exhaustive guide to AMM's'

[14] Finder. 'Automated Market Makers'

[15] The Defiant 'What are automated market makers like Uniswap?'

[16] Gemini 'Flash loans'

[17] Coindesk 'What is a flash loan?'

[18] The Defiant 'Flash loans'

[19] CMC 'What are flash loan attacks?'

[20] Bloomberg 'Flash loans are providing instant cash to crypto speculators'

[21] Kalinoffblog 'Flash loans Ethereum'

[22] Liquidblog 'Flash loan defi'

[23] Financial Times. 'What are cryptocurrencies and stablecoins and how do they work?'

[24] Coindesk. 'What are stable coins?'

[25] Grant Bartel. 'What is the point of a stablecoin?'

[26] Coinbase. 'What is a stablecoin?'

[27] Coindesk. 'MakerDAO's Embrace of Centralized Stablecoins Offers Risks and Rewards'

[28] Investopedia. 'Stablecoins'

[29] US News. 'What are stablecoins and how can I invest in them?'

[30] Corporate Finance Institute. 'Stablecoins'

[31] Flexiple. 'What are stablecoins and how do they work?'

[32] FTX. 'What are futures?'

[33] Binance. 'What are perpetual futures contracts?'

[34] Medium. 'A quick look into perpetual futures contracts'

[35] BitMex. 'Perpetual contracts guide'

[36] Eqonex. 'Perpetuals trading series, what are perpetual futures and why trade them'

[37] Interactive Crypto. ’Perpetual future contracts’

[38] youtube. ’REST API’

[39] youtube. ’Understanding API’s’

[40] FTX. ’API dox’

[41] Github. ’ccxt’

[42] Github. ’ccxt - help 1’

[43] Github. ’ccxt - help 2’

[44] FTX. ’Get historical index’

[45] FTX. ’Markets’

[46] ccxt. ’Symbols and market ids’

[47] Geneseo. ’Risk Vs Return’

[48] Youtube. ’Sharpe Ration: Key metrics when building a bot’

[49] Youtube. ’Sharpe vs Sortino’

[50] Groww. ’Tracking error’

[51] Github. ’Sandbox exchange’