HFT portfolio

Aarhus University

August 2023

This report contains 5495 words which amounts to approximately 31459 characters.

Question 1

Using the market for lemons example explain why information asymmetry can cause market failure. Further, discuss how this basic theory can be used in the general features of a quote-driven dealer market and the associated risks in the context of High Frequency Finance.

The "Lemons" problem arises because of asymmetric information held by the buyer and the seller of a product. The theory can be explained using the used car market. In this market the buyer does not know the state of the car (if it is a bad car = lemon or a premium car = peach). On the contrary the seller knows the state of the car. This results in asymmetric information between the buyer and seller [1].
Now assume that the price of a lemon is: $p_l = \underline{V}$ and that the price of a peach is: $p_p = \overline{V}$. Under these assumptions we have that $p_l < p_p$.

The buyer knows that there is a probability that a car is a lemon. Therefore, the buyer does not want to pay more than the average price: $E(V) = \lambda \cdot \underline{V} + (1 - \lambda) \cdot \overline{V}$. Here the λ is the proportion of lemons on the market.
This creates a problematic situation, since $E(V) > p_l$ and $E(V) < p_p$. The market therefore generates a dynamic that awards the sellers of lemons [2]. Because the lemon can be sold for an overprice in the market. On the other side, no seller of a peach is willing to sell at the average price, since this price is below the value of the car. Therefore these market conditions drive adverse selection. This can be seen by the fact that sellers gain profit by selling a lemon instead of optimizing for selling cars for the fair price. This will disturb the market's equilibrium and therefore lead to market failure [2, 3].

The lemon theory shows that the price of an asset can be influenced by the information a trader has of the asset. Additionally if the bid-ask spread is wide, then it can be difficult to trade at the "fair-price" [4]. The size of the spread is a sign of the uncertainty of the fair-price. The spread will also increase if some traders has important information that other traders don't [5, page 74]. This situation creates the exact lemon scenario. This is because the "informed" traders make money on their information while "uninformed" traders are punished by the increased bid-ask spread. These market condition drive adverse selection which can lead to market failure [5, page 74].

On one hand an argument could be made, that a broker in a quote-driven dealer market should direct a trader to the dealer with the narrowest bid-ask spread. This will ensure that the trader most likely would trade close to the fair-price. On the other hand, one could argue that the broker should direct a trader to the dealer offering the best price for the position the trader intends to take. For instance, if a trader wishes to buy, then he would prefer the lowest possible ask-price. The first argument generate most trades for the dealers with the most narrow bid-ask spread. The second argument generates most trades for the dealers who offer the best price. The dealer with the narrowest bid-ask spread will probably also offer the best bid-ask prices.

For most high frequency finance, the goal is to generalize to the true signal (price) of the market under normal circumstances [6]. It is generally advised against using HFT[1] strategies during economical events (anomalous market conditions). As discussed in the lectures, a HFT strategy should not execute trades in the time gab (at minimum):

[1]HFT = High Frequency Trading

30 minutes before and 20 minutes after an event. An example of an event could be a change to the interest rate. A sudden event can cause information asymmetry, which can lead to a less-than-ideal scenario as discussed by the lemons example.

On one hand an argument could be made that HFT strategies should be turned off during such events (if possible). This argument builds on the idea that the HFT strategy does not have the information that drives the price (during the event). For example, a HFT strategy that receive only market data (price, volatility and volume) would be at a disadvantage. In such a scenario the information of an event might not be encoded in the prices observed (especially if the event happens suddenly). This makes the strategy an "unaware" trader during the event, which creates an asymmetric information scenario. This will most likely result in less-than-ideal trades (as can be seen by being a buyer in the lemon-scenario). On the other hand, an argument could be made, that a HFT strategy should not be turned off. Especially if it is possible to process the event information in the HFT strategy. This argument build on the idea, that the "first-movers", which are the first to trade this new information, will gain an advantage. This is because of the possible information asymmetry as the event is being realized in the markets. This would equate to being a seller in the lemon-scenario. Another compelling argument for using HFT strategies during events, would be that these strategies does not have "human-feelings". Meaning that such strategies would not face behavioural biases, and therefore hopefully not enter into losing trades [6].

Question 2

Consider a high frequency trading platform of your choice and explain how one can remove micro-structure noise? Download the data at different frequencies and do basic analysis to identify any existing stylized facts.

One goal in parameter/measure estimation is to make the estimate as precise as possible. This is often achieved by using all available data. This creates an interest in estimating empirical measures or parameters using tick-by-tick data. Estimation in live systems though prove difficult due to the introduction of bias in the form of micro-structure noise. This noise, from e.g. bounces in the bid- and ask-prices, has proven to introduce significant bias in some high-frequency parameters. The noise of bid-ask bounces can also add bias in the volatility estimates [7, page 60]. The introduced noise suggests a bias-variance compromise when choosing the sampling frequency. A high frequency introduces more data but also bias, whereas a lower frequency removes some bias, but less data is available. Some modelers have e.g. opted for a granularity of 5 minutes instead of tick-by-tick data [8, page 2].
Market micro-structure noise consists of different frictions in the trading process. Examples of this could be: *bid-ask bounces, discreteness of price changes, differences in trade sizes or informational content of price changes, etc* [9, page 3].

The following formulation can be used to describe the log-price process:

$$Y_t = X_t + \epsilon_t$$

Here Y_t is the observed log-price in the market. The Y_t is a combination of the unobserved efficient log-price X_t and a noise term: ϵ_t [9, page 4]. Here the ϵ_t represents the micro-structure noise that we want to remove.
Instead of using a more moderate frequency one can use techniques for bias correction in the tick-by-tick data. For instance one can use techniques such as a weighted mid-price or a micro price (calculated from the bid- and ask-price) [10]. More advanced techniques include filtering in forms of moving average filtering or autoregressive filtering. Other techniques such as a sub-sample approach or a kernel-based approach can also be implemented for bias correction [8, page 2].

Certain models that represent price dynamics, might include a noise-term. If the micro-structure noise is not removed prior to the model-fitting, then the noise would be captured in this term.
Examples are the stochastic volatility (SV) models. Assume that log-returns are calculated as $x_t = log(\frac{p_t}{p_{t-1}})$ and that these follow the distribution:

$$x_t = \epsilon_t \cdot \beta \cdot exp(g_t/2), \quad g_{t+1} = \phi g_t + \eta_t$$

Where $|\phi| < 1$ and it is assumes that $\{\epsilon_t\}$ and $\{\eta_t\}$ are independent. It is also assumed that $\epsilon_t \sim N(0, 1)$ and $\eta_t \sim N(0, \sigma^2)$ [11, page 262-263]. In such a model the micro-structure noise would be "absorbed" in the ϵ_t term. One should though note that this model might not be suited for tick-by-tick data. The reason is that the underlying model (g_{t+1}) follows an auto-regressive (AR) process. The AR process assume that data is observed at equal spaced time-intervals. Lunde & Hansen (2006) reported that the micro-structure noise has some ugly facts such as being correlated with the efficient price [8, page 1]. A possible way to deal with such a correlation, is to extend the model. For instance in the SV model it is possible to make the two error-terms: $\{\epsilon_t\}$ and $\{\eta_t\}$ correlated. This would give feedback from previous returns to the future returns [11, page 265-266], and make a correlation between the unobserved price and the noise. This can be achieved by assuming that:

$$\begin{pmatrix} \varepsilon_t \\ \eta_t \end{pmatrix} \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma}), \ \text{ with } \mathbf{\Sigma} = \begin{pmatrix} 1 & \rho\sigma \\ \rho\sigma & \sigma^2 \end{pmatrix}$$

Note that this model is not necessarily meant to deal with the micro-structure noise, but the terms included in the model allow for some noise, and correlation of noises. If the noise is very significant and has extreme properties (as noted by Lunde & Hansen) then this type of model might not be optimal to use. In that case (and most other cases) it would be best to deal with the micro-structure noise prior to model-fitting.

For this assignment the cTrader platform has been used to download tick-by-tick data. This platform only supports open-prices for minute/hour/day etc. data. The tick data has therefore been down sampled to minute, hour and daily prices in Python. The code is available in the following Github-link. Stylized facts are certain empirical statistical properties that might be observed in different financial markets over different time-periods [12, page 32].

In this report Bitcoin to us dollars data has been considered. The data is from in the period: 2023-06-27 to 2023-07-27. The bid-ask spread is first considered:

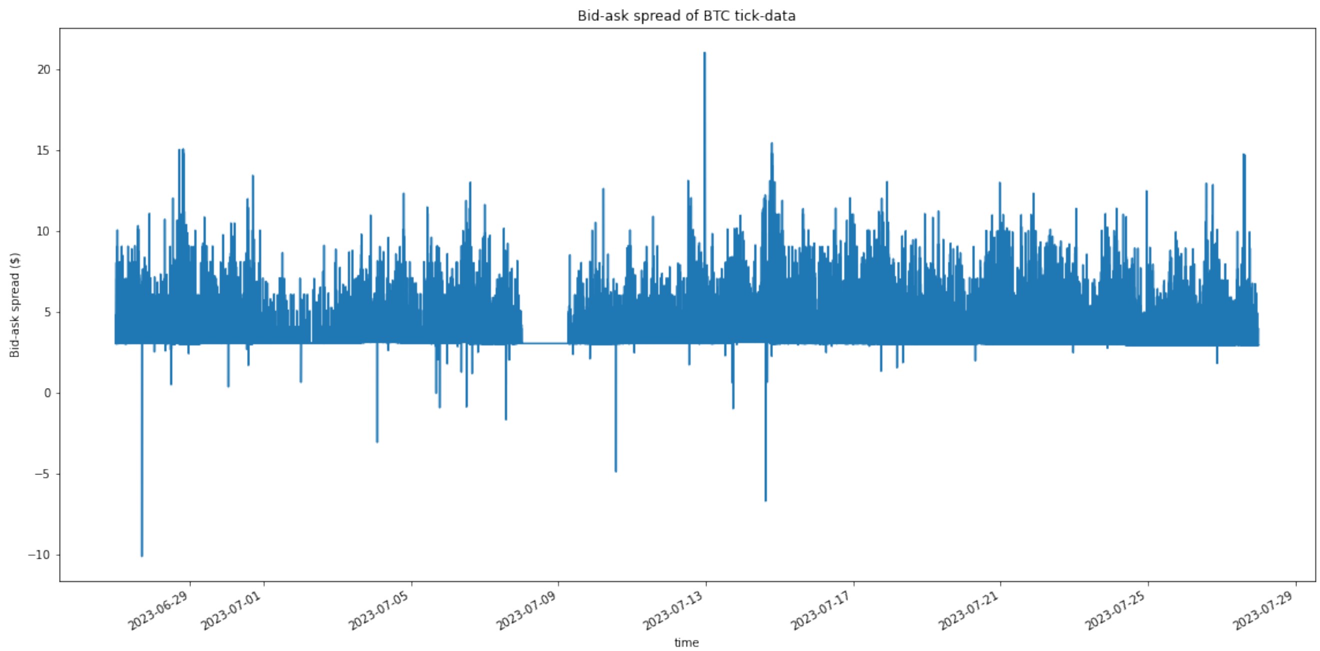

Figure 1: Bid-ask spread of tick data

This plot illustrate that the bid-ask spread varies over time, and that some mistake (e.g. downtime) has occurred around 2023-07-09. It can also be seen that sometimes the spread is negative. This should not be possible, otherwise there would be quite an obvious arbitrage opportunity. The negative spreads might stem from micro-structure noise or wrong recording of the price by the platform. To eliminate micro-structure noise in the data, the mid-price between ask and bid has been used for the analysis. In the analysis the log-return[2] is considered, which can be calculated by: $x_t = log(\frac{p_t}{p_{t-1}})$.

[2] log-returns and returns will be used interchangeably unless specified otherwise.

Fact 1: Distribution of returns

These plots illustrate the distribution of the log-returns using the tick-by-tick data:

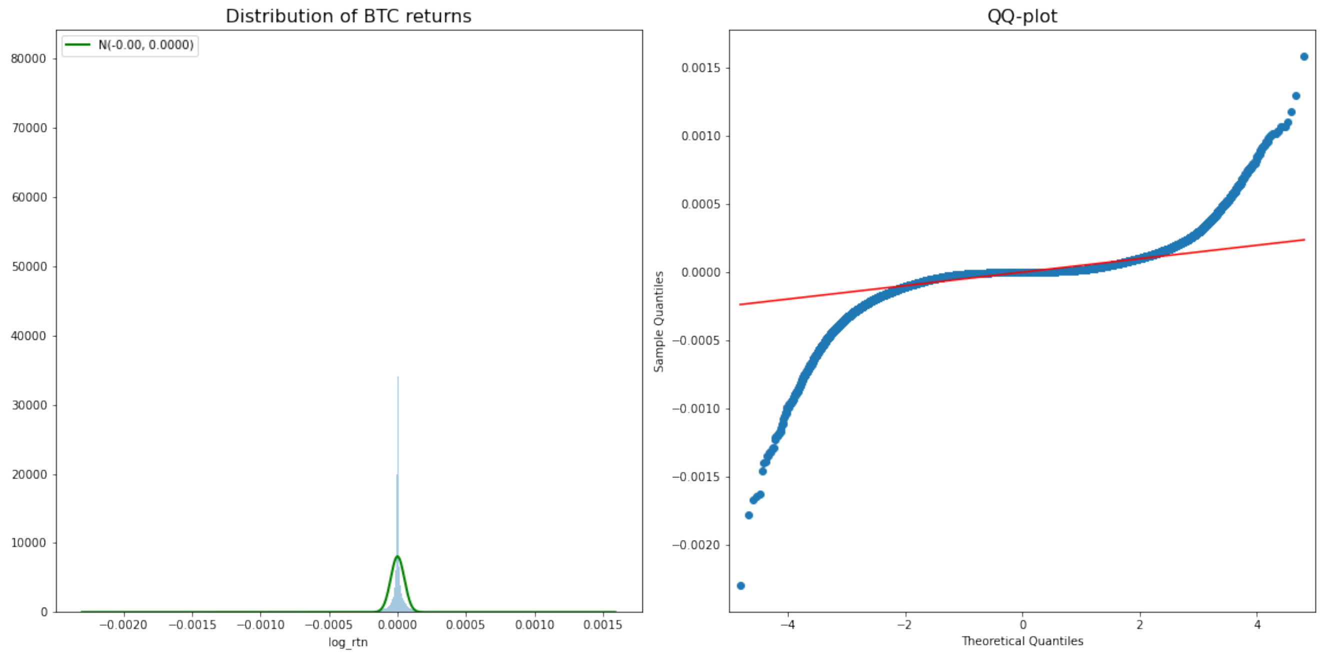

Figure 2: Histogram and QQ-plot of tick-by-tick returns

Descriptive Statistics:
Range of dates: 2023-06-27 - 2023-07-27
Number of observations: 1306778
Mean: -0.0000
Median: 0.0000
Min: -0.0023
Max: 0.0016
Standard Deviation: 0.0000
Skewness: -1.1467
Kurtosis: 56.2905
Jarque-Bera statistic: 172813289.00 with p-value: 0.00

These plots illustrate the distribution of the log-returns using the per minute data:

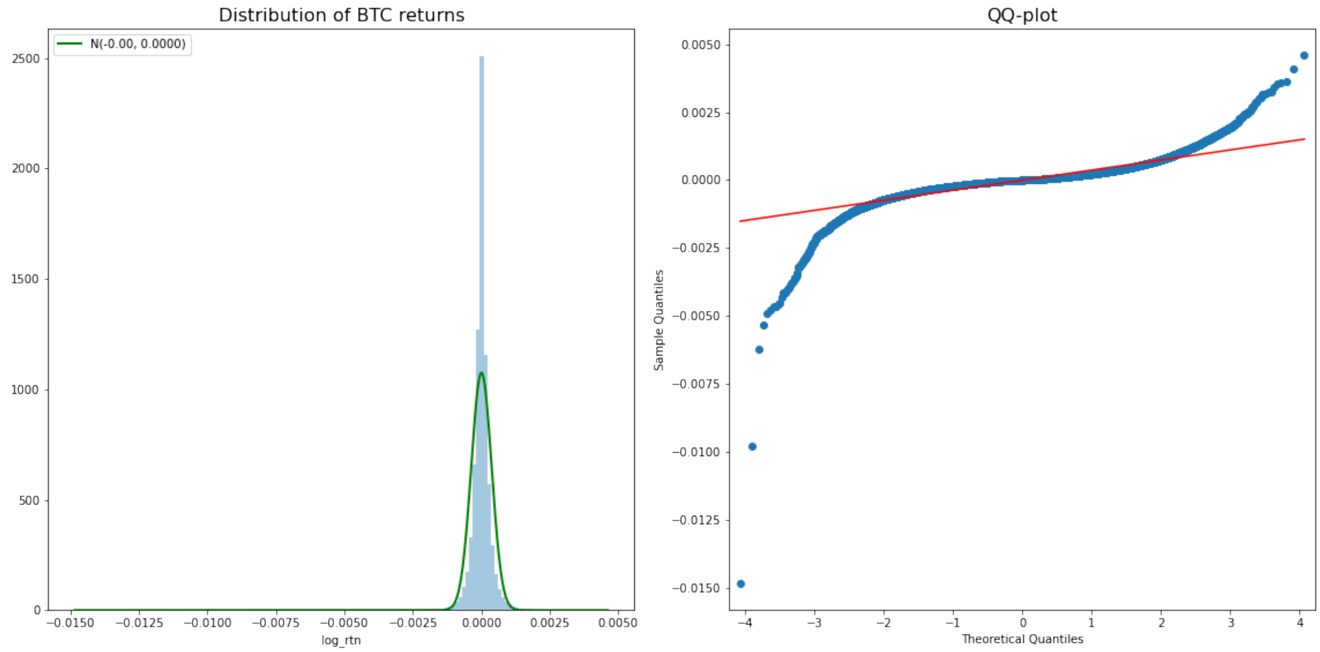

Figure 3: Histogram and QQ-plot of minute returns

Descriptive Statistics:
Range of dates: 2023-06-27 - 2023-07-27
Number of observations: 41979
Mean: -0.0000
Median: -0.0000
Min: -0.0148
Max: 0.0046
Standard Deviation: 0.0004
Skewness: -2.6426
Kurtosis: 92.4248
Jarque-Bera statistic: 14986862.45 with p-value: 0.00

These plots illustrate the distribution of the log-returns using the per hour data:

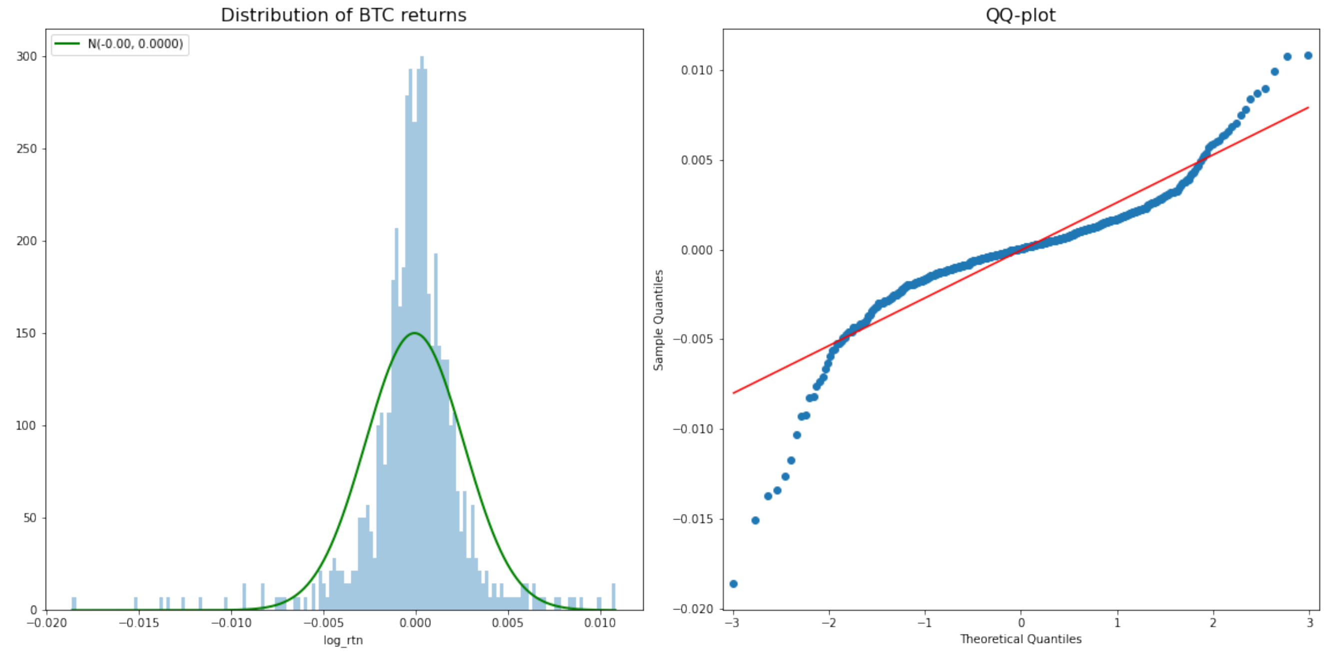

Figure 4: Histogram and QQ-plot of hourly returns

Descriptive Statistics:
Range of dates: 2023-06-27 - 2023-07-27
Number of observations: 713
Mean: -0.0000
Median: 0.0001
Min: -0.0186
Max: 0.0108
Standard Deviation: 0.0027
Skewness: -1.1537
Kurtosis: 9.1640
Jarque-Bera statistic: 2613.02 with p-value: 0.00

These plots illustrate the distribution of the log-returns using the per day data:

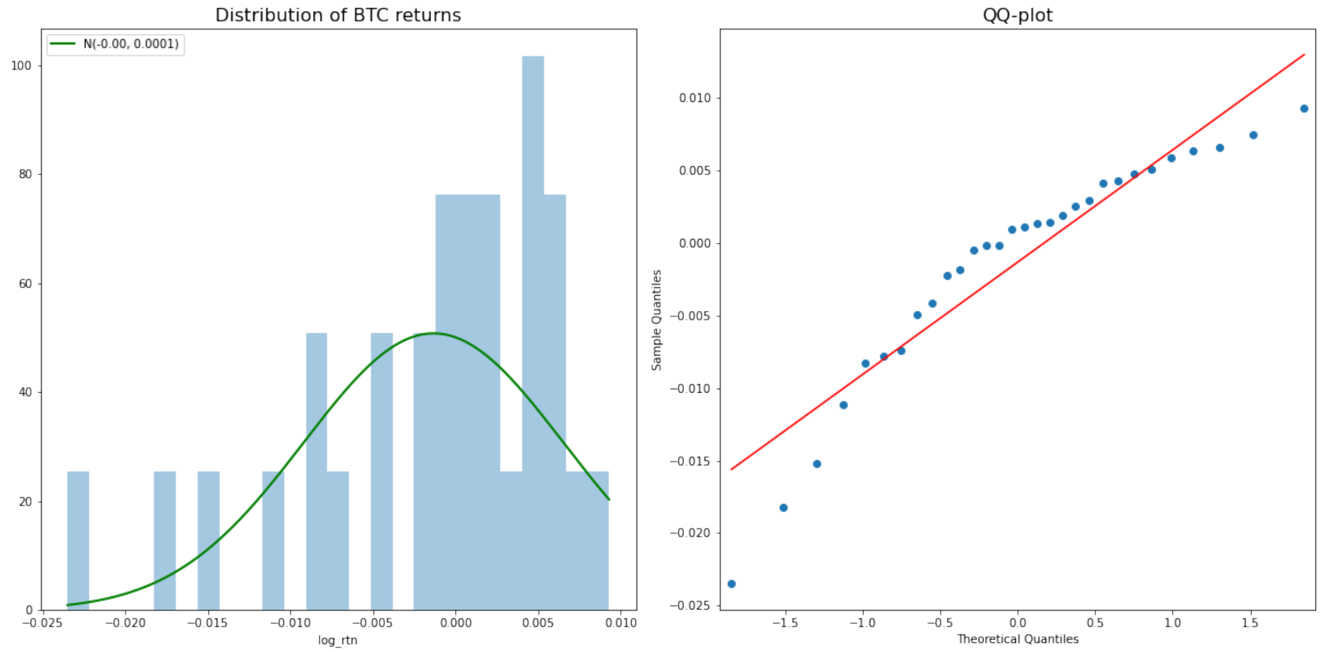

Figure 5: Histogram and QQ-plot of daily returns

Descriptive Statistics:
Range of dates: 2023-06-28 - 2023-07-27
Number of observations: 30
Mean: -0.0013
Median: 0.0010
Min: -0.0235
Max: 0.0093
Standard Deviation: 0.0079
Skewness: -1.2154
Kurtosis: 1.2154
Jarque-Bera statistic: 7.51 with p-value: 0.02

All plots indicate that the Gaussian-distribution is inadequate for explaining the distribution of the log-returns. It seems that the distribution of log-returns has to many observations around 0 and too fat-tails for a normal distribution. These observations are supported by the excess kurtosis observed in the statistics summary. The dots in the QQ-plot also deviate too much from the actual theoretical line to be normal-distributed. In the summary-statics it is observed that the Jarque-Bera statistic is very large. This produce a very low p-value which imply that the null-hypothesis is to be rejected. The null-hypothesis assumes that the skewness and kurtosis match a normal-distribution. From this it can be concluded that returns are non-Gaussian.
From the statistics a negative skewness is observed. This means that returns that are large and negative is more often seen than large positive returns (skewing the distribution towards negative values).

Fact 2: Autocorrelation in returns

The following plots display the autocorrelation of returns, squared returns and absolute returns.

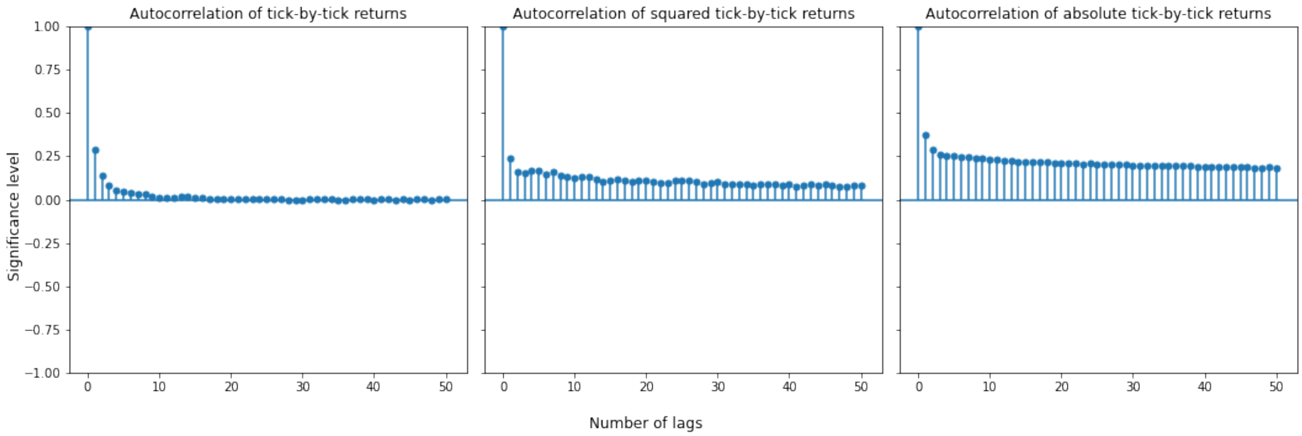

Figure 6: ACF-plots of tick-by-tick log-returns

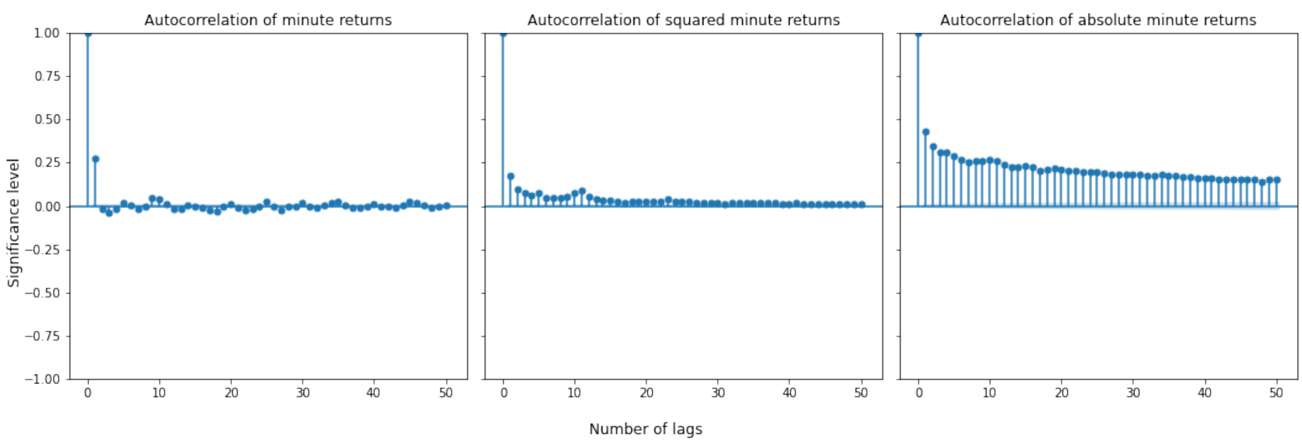

Figure 7: ACF-plots of minute log-returns

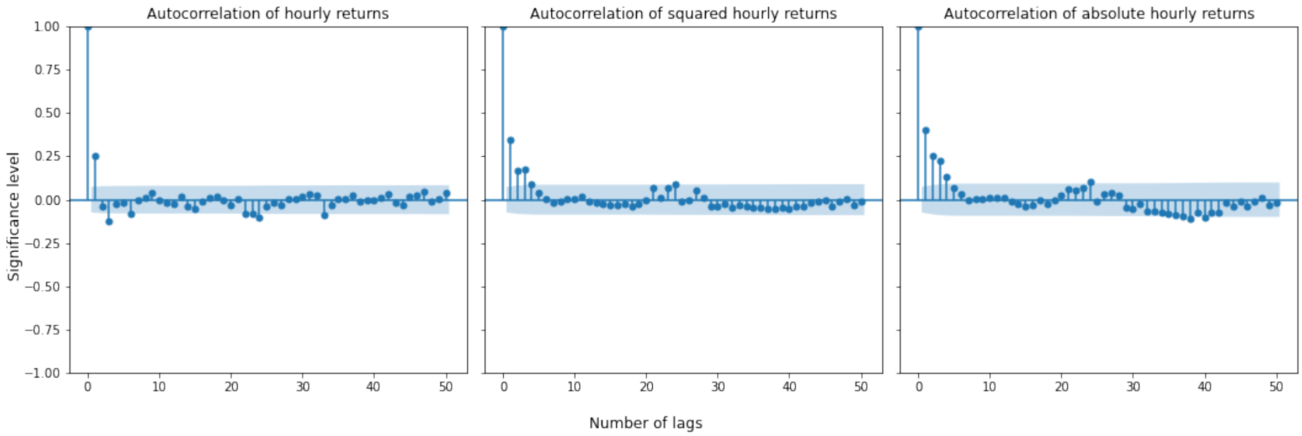

Figure 8: ACF-plots of hourly log-returns

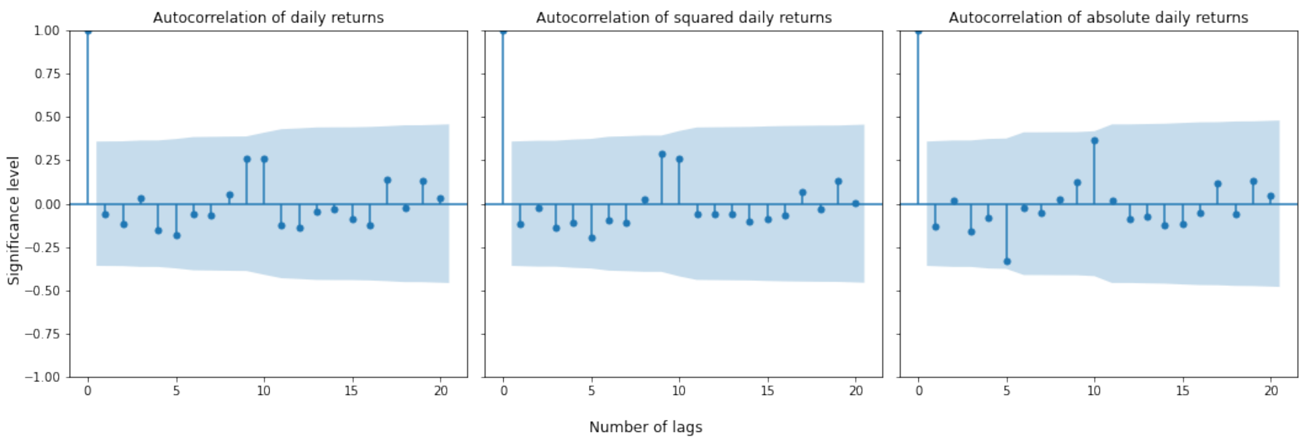

Figure 9: ACF-plots of daily log-returns

The ACF-plots of the log-returns display that correlation over time is mostly insignificant. Though at some frequencies the lag 1 display some significance. It is important that one does not confuse a correlation of zero with independence. In general:

$$cov(r_t, r_{t-1}) = 0 \;\not\Longrightarrow\; r_t \perp\!\!\!\perp r_{t-1}$$

From the absolute and squared log-returns there are some dependence of past log-returns (for the tick, minute and hour data). This is because there is significant observations for the different lags. This means that the size of the return is correlated over time, but the direction of the return is not (if the price goes up or down). The daily and hourly data show less significance (in all ACF-plots) over time than the minute and tick data. This indicates that log-returns on lower granularity data is less dependent on past log-returns. It should be noted that there are fewer data-points for the hour and day-data, which can have an effect. It should also be noted that the ACF-function is a time-series analysis method, which assume that data arrive at equal-spaced intervals. This is not the case for the tick-by-tick data.

Fact 3: Volatility clustering (changing volatility)

The following plots display the asset-price, the log return and volatility. On the volatility plot is a 50 and 10 period rolling volatility measure (standard deviation).

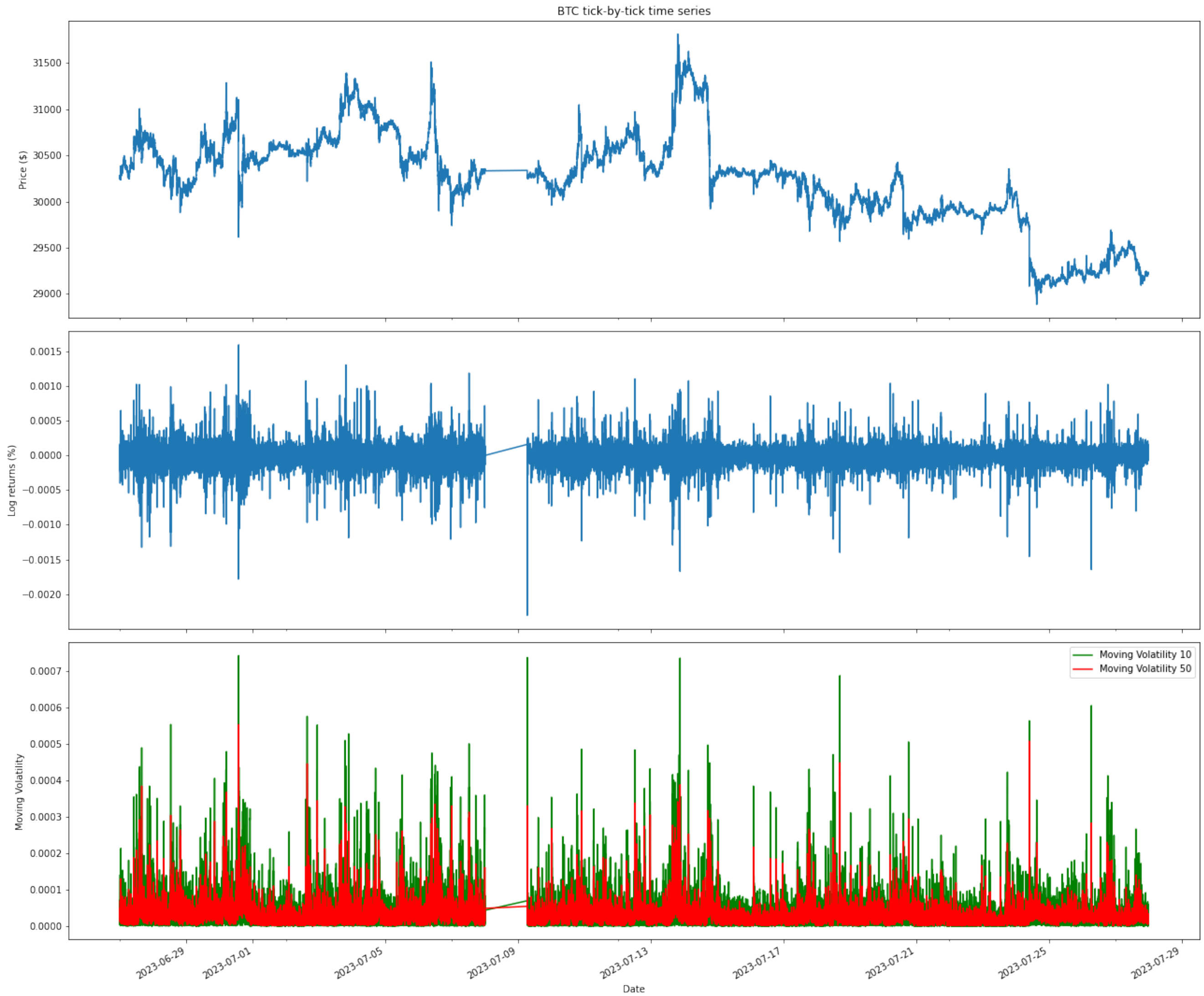

Figure 10: Price, log-return and volatility plot of tick data

Figure 11: Price, log-return and volatility plot of per minute data

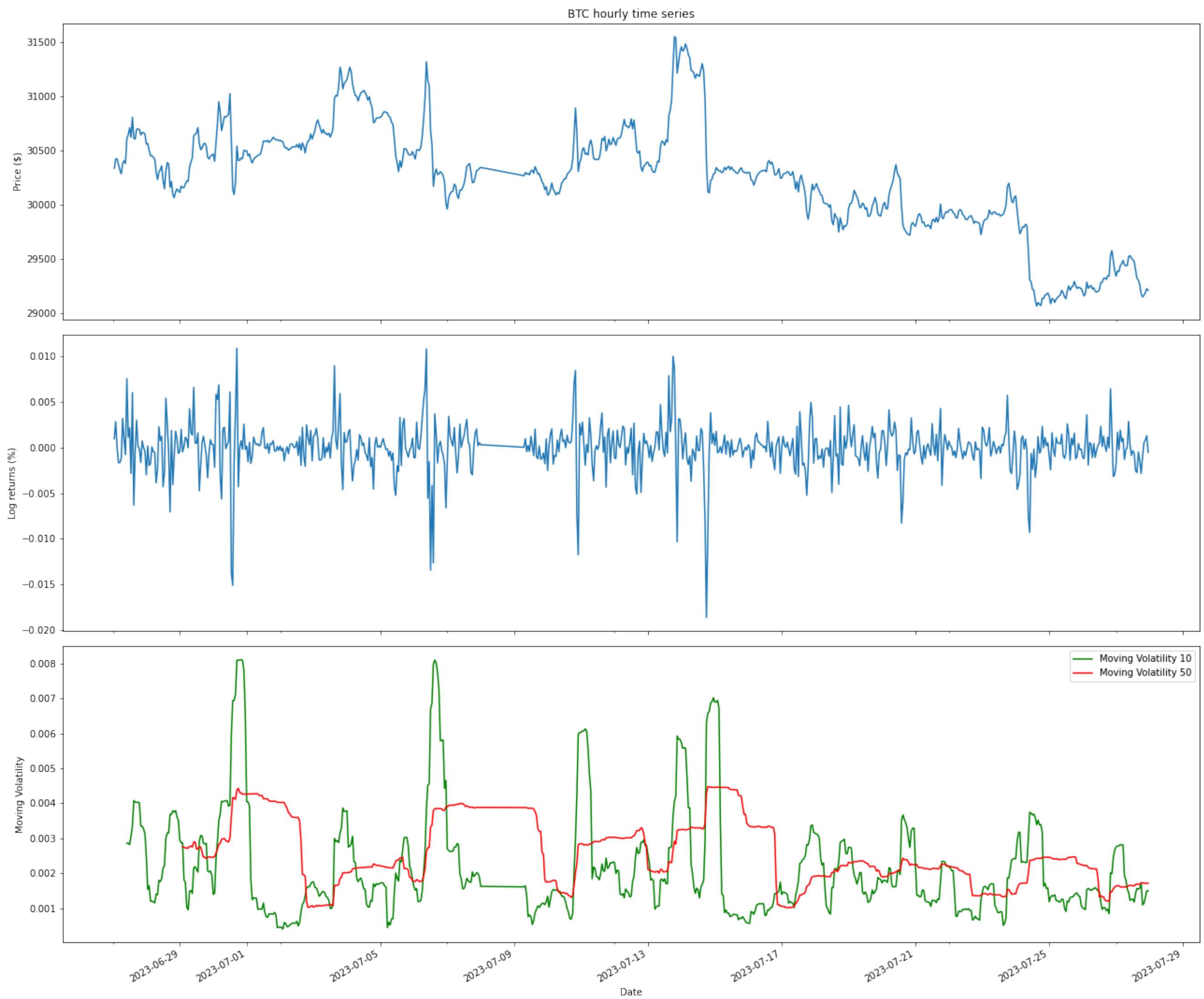

Figure 12: Price, log-return and volatility plot of hourly data

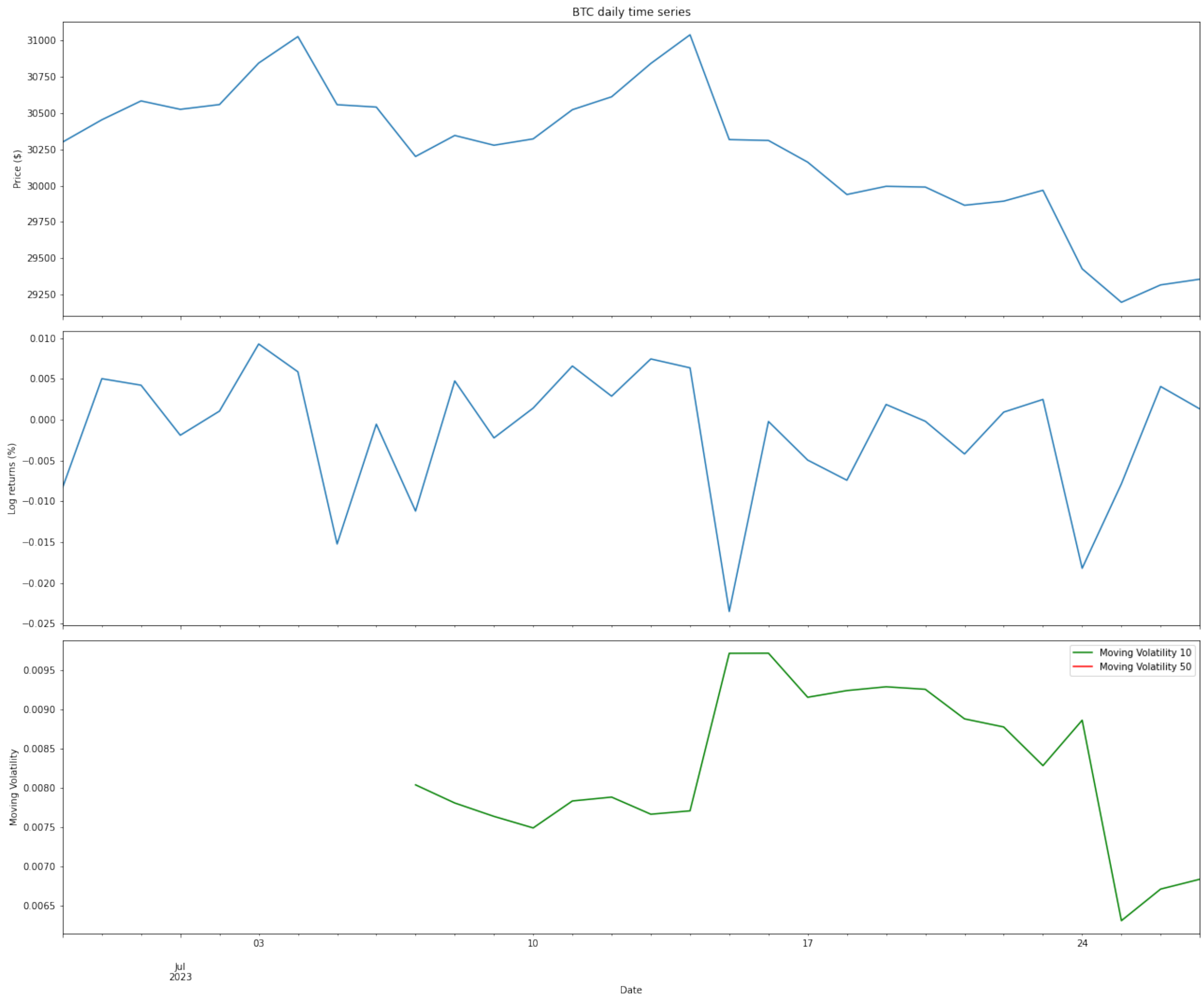

Figure 13: Price, log-return and volatility plot of daily data

From the different plots it can be observed that the volatility changes over time. The volatility is sometimes clustered together in highly volatile periods (see clusters on the log-return plots). The increasing and decreasing volatility can also be seen by the rolling volatility measures for all the different time-intervals. There are only 30 daily observations, so the change of volatility is not as obvious on the plot for this time-interval.

Stylized facts are important as quality measures. Models with the goal of representing asset price dynamics should be able to replicate these stylized facts [12, page 32].

Question 3

Explain scale invariant patterns in financial markets. An algorithmic trader would like to use scale invariant patterns in the data from FX market. Propose a trading strategy (only pseudo code or steps that are involved in the trading strategy) for high frequency trading using any scaling law of your choice. Explain to the trader the logic behind your trading strategy.

A scale invariant function fulfills the condition:

$$f(\lambda x) = \lambda^{\Delta} f(x)$$

This allow the function to work on different time scales (by re-scaling the x) [13]. The power-laws: $f(x) = \alpha \cdot x^{\beta}$ are examples of scale invariant functions. Power-laws have been observed in both economical and financial data [14, page 2]. Glattfelder, Dupuis & Olsen (2011) provide 12 empirical scaling laws that work on financial data [15].

There are different reasons for using scale invariant functions for algorithmic trading. It is observed that the strategy can be used on different frequencies of data (tick, minute, hour, etc.). Scale invariant functions can avoid problems created by interval based summery of prices. The interval structure might miss some important moves in the market because events might happen in-between the intervals. An approach is using certain events (e.g. a directional change), and rely on these events instead of time [16]. The directional changes can be explained by the plot:

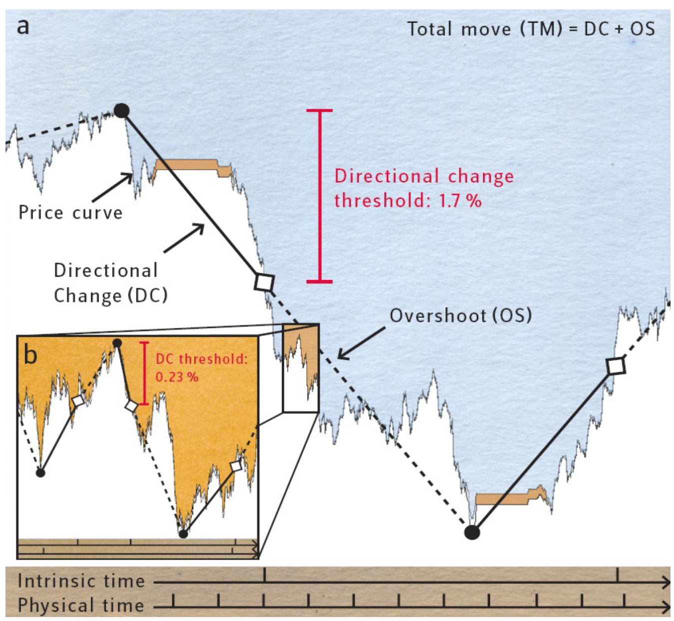

Figure 14: Display of directional changes, taken from: [16]

This plot illustrates the total moves (TM) of the strategy (which is from a solid point to a solid point). The total move consist of a directional change (DC) and an overshoot (OS). The only parameter is the directional change: Δx_{dc}. This parameter is a threshold measured in the percent change of the underlying asset. When the percent change has dropped by Δx_{dc} a down-ward trend has been triggered (and opposite for the up-ward trend). This strategy alternates between up-ward and down-ward trends. The overshoot value is found between the event (white diamond) and the start of the new event (black dot) [16]. In Figure 14 the directional change was set to 1.7 % in plot a, and 0.23 % in plot b. Comparing the plots it is found that the larger Δx_{dc}, the fewer events are triggered. The amount of directional changes can be modeled by the power-law [16]:

$$N(\Delta x_{dc}) = \left(\frac{\Delta x_{dc}}{C_{N,dc}}\right)^{E_{N,dc}}$$

The following definition for the mean value is used [15, page 2]:

$$\langle x \rangle_p = (\frac{1}{n}\sum_{i=1}^{n}(x_i^p))^{\frac{1}{p}}, \quad p \in \{1, 2\}$$

The scaling law for the average overshoot value is a function of the directional-change threshold Δx_{dc}. The power-law function can be expressed as [15, page 7]:

$$\langle \Delta x^{OS} \rangle_1 = (\frac{\Delta x_{dc}}{C_{x,OS}})^{E_{x,OS}}$$

The different parameters can be estimated by using the logarithm and fitting a linear function:

$$log(\langle \Delta x^{OS} \rangle_1) = E_{x,OS} \cdot log(\frac{1}{C_{x,OS}}) + E_{x,OS} \cdot log(\Delta x_{dc})$$

The observational pair (Δx_{dc}, $\langle \Delta x^{OS} \rangle_1$) can be estimated using empirical analysis of price data. This can be done for multiple values of Δx_{dc}. This will generate multiple data-points to which a function can be fitted.
The following pseudo-code explains how empirical observations can be saved for a directional change strategy [15, page 12]:

Algorithm 1 DC(price, Δx_{dc})

$Initialize\ (only\ for\ first\ iteration) : direction = 0,\ x^{ext} = price$
if $direction == 0$ **then** ▷ Only used for starting the algorithm
  if $x^{ext} < price$ **then**
    $x^{ext} = price$
  else
    $diff = (price - x^{ext})/x^{ext}$
    if $diff \leq -\Delta x_{dc}$ **then**
      $direction = -1$
      $x^{ext} = price$
      $event\_price = price$
else if $direction == 1$ **then**
  if $x^{ext} < price$ **then**
    $x^{ext} = price$
  else
    $diff = (price - x^{ext})/x^{ext}$
    if $diff \leq -\Delta x_{dc}$ **then**
      $direction = -1$
      $OSV = (x^{ext} - event\_price)/event\_price$
      $x^{ext} = price$
      $event\_price = price$
      $Write\ OSV\ to\ file$ ▷ Write to "up-trend" file
else if $direction == -1$ **then**
  if $x^{ext} > price$ **then**
    $x^{ext} = price$
  else
    $diff = (price - x^{ext})/x^{ext}$
    if $diff \geq \Delta x_{dc}$ **then**
      $direction = 1$
      $OSV = (event\_price - x^{ext})/event\_price$
      $x^{ext} = price$
      $event\_price = price$
      $Write\ OSV\ to\ file$ ▷ Write to "down-trend" file

The code assume that the function is used as a generator (in Python sense). This means that the function is run after each new update to the price. This code has been implemented in C#, so that it can be used in cTrader. See

the code in this Github-link. The code is run with values of Δx_{dc} from 0.1% to 7% with 0.1% increments. The last five years tick-by-tick Bitcoin data was used (in cTrader).
In the following plots, the average overshoot value $\langle \Delta x^{OS} \rangle_1 = \langle OSV \rangle_1$ is related to the directional change parameter Δx_{dc}. The following analysis can be found in this Github-link. In the down-loading process the overshoot value is considered for up-trends and down-trends only (see comments in Algorithm 1).
These plots consider up-trend data only:

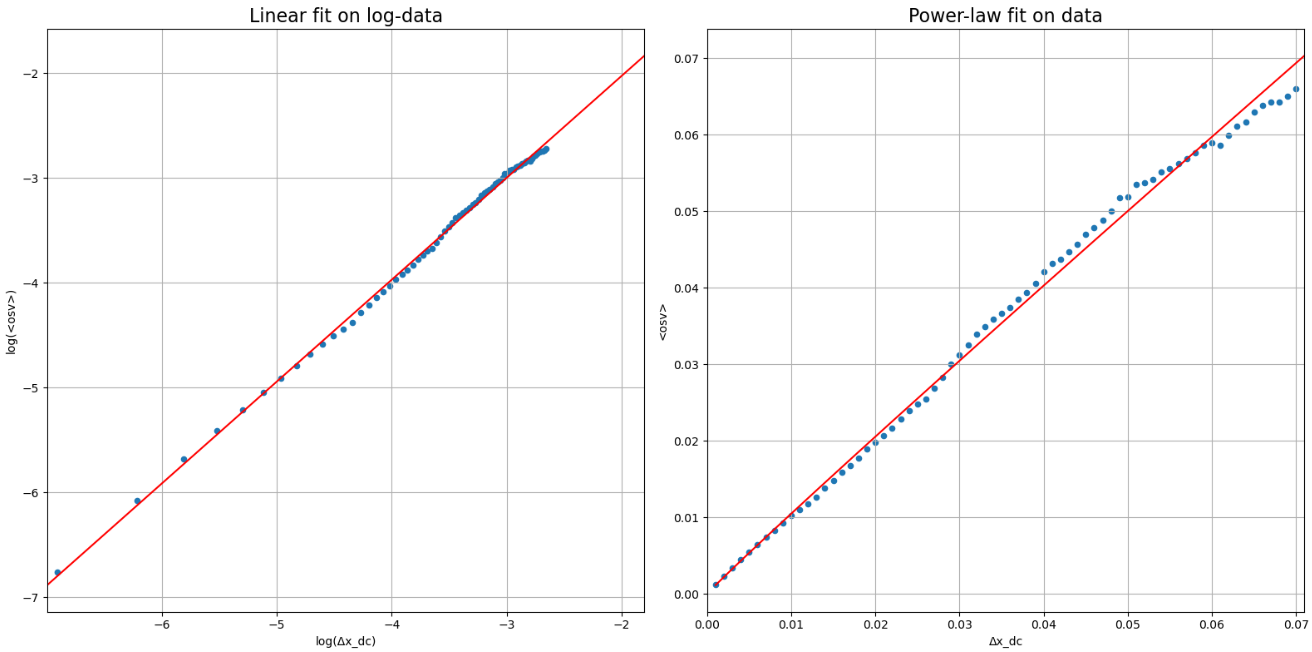

Figure 15: Power-law fitting of $\langle OSV \rangle_1$ for up-trend data

The plots display that the power-law capture the underlying trend of data. Though the fit seems a bit worse the larger Δx_{dc} become. It should be noted that for larger Δx_{dc} less observations are made, and therefore a bit of noise is expected.
From the power-law the variables are found to be: $E_{x,OS} = 0.972 \pm 0.010$ and $C_{x,OS} = 1.090$. With these variables the function becomes:

$$\langle \Delta x^{OS} \rangle_1 = (\frac{\Delta x_{dc}}{1.090})^{0.972}$$

The following plots display the function fitting to the down-trend data:

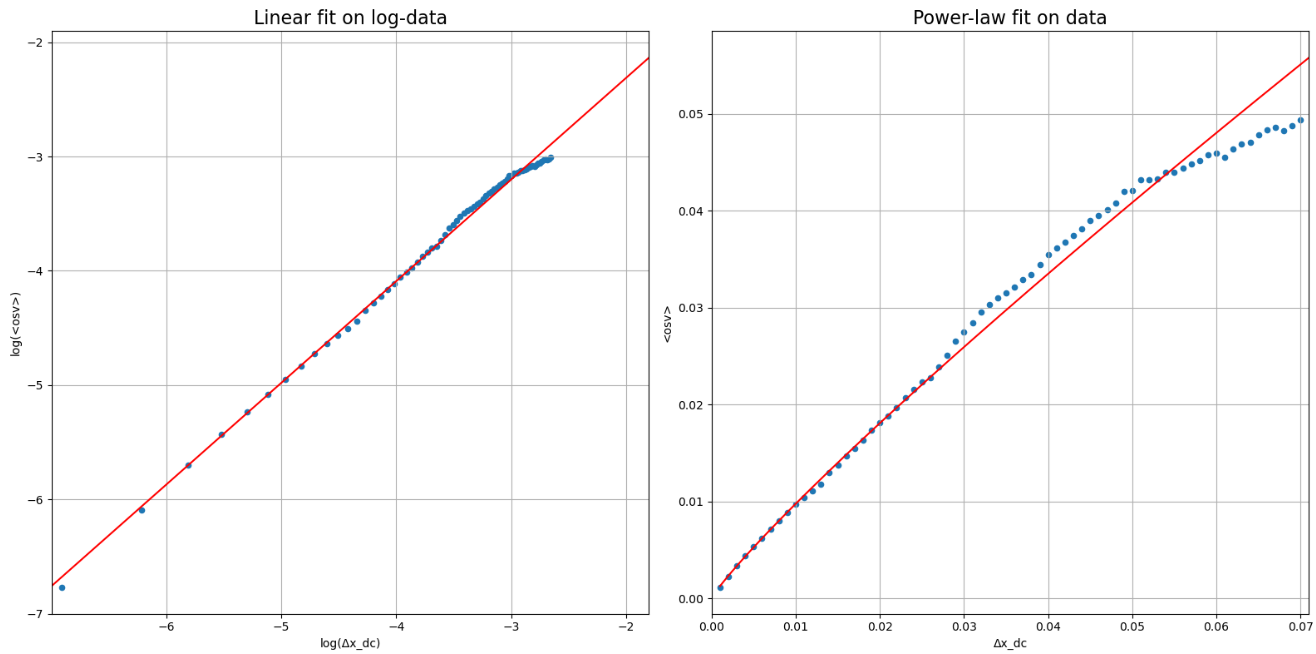

Figure 16: Power-law fitting of $\langle OSV \rangle_1$ for down-trend data

The fit to the down-trend data seems appropriate. It is though observed that the larger values of Δx_{dc} deviate a bit from the fit. The parameters are estimated to: $E_{x,OS} = 0.889 \pm 0.013$ and $C_{x,OS} = 1.820$. This yield the equation:

$$\langle \Delta x^{OS} \rangle_1 = (\frac{\Delta x_{dc}}{1.820})^{0.889} \quad \text{(OS\_short\_values)}$$

In the next plots the models are fitted to data for both up- and down-trend events:

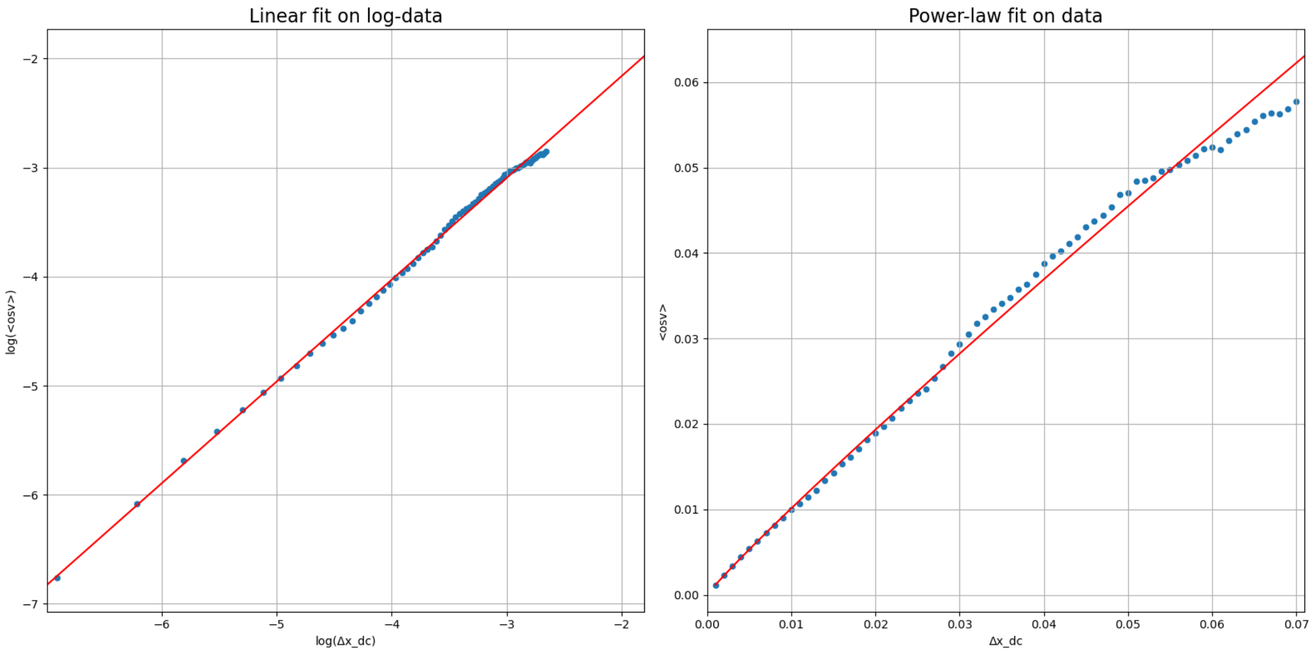

Figure 17: Power-law fitting of $\langle OSV \rangle_1$

The power-law seems to fit the data. For this model the estimates are: $E_{x,OS} = 0.933 \pm 0.010$ and $C_{x,OS} = 1.371$. This can be written as:

$$\langle \Delta x^{OS} \rangle_1 = (\frac{\Delta x_{dc}}{1.371})^{0.933}$$

The following plots relate the amount of directional changes to the threshold Δx_{dc}:

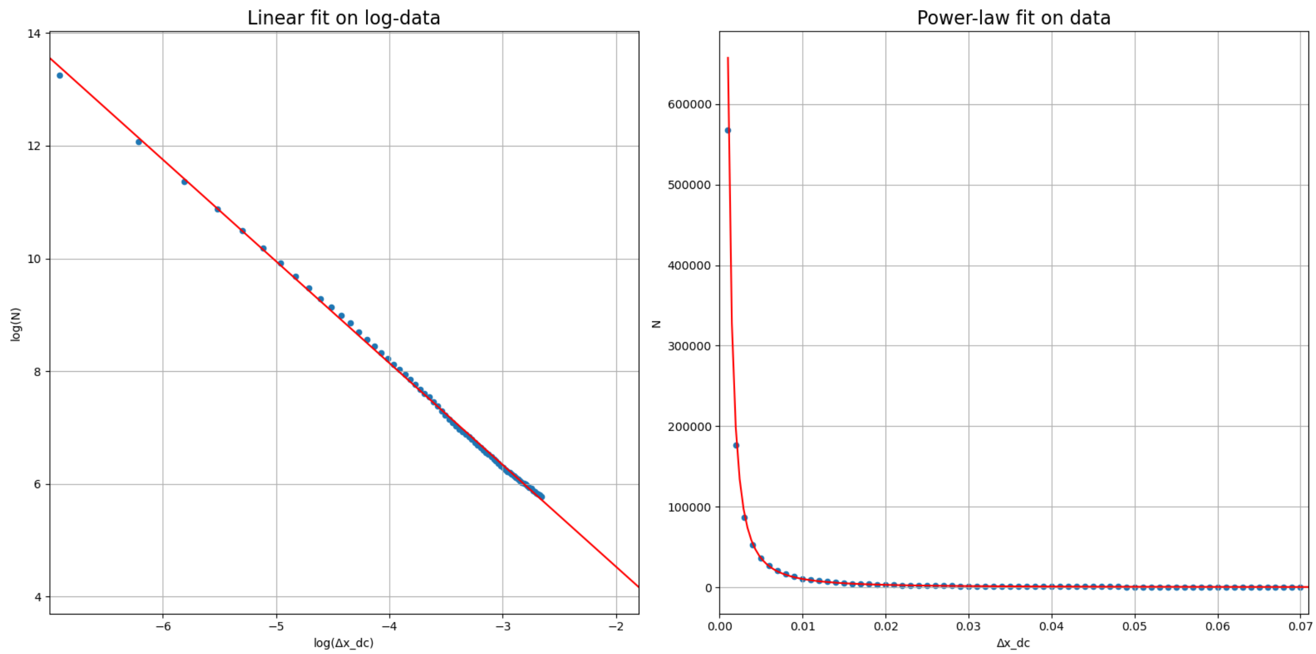

Figure 18: Power-law fitting of $N(\Delta x_{dc})$

The plots illustrate a negative slope and thus a decaying power-law. It makes sense, that as Δx_{dc} increases, then fewer observations are seen. Meaning that it takes longer to realize larger changes (as observed in Figure 14). From the fit it is found that $E_{N,de} = -1.807 \pm 0.012$ and that $C_{N,de} = 1.661$ making the power-law equal to:

$$N(\Delta x^{OS}) = (\frac{\Delta x_{dc}}{1.661})^{-1.807}$$

Since the up- and down-trends alternates, it is sufficient to use data from one trend only (the difference of observations can't be more than 1).

With this in mind a trading strategy can be formed and explained to the trader. In this trading-strategy a short position is entered after a down-ward event has occurred. A parameter Δx_{exit} is created to determine when to close the short position. Δx_{exit} is how many percentage the asset should drop in price before the strategy close the short-position. The Δx_{exit} imitates a take-profit order. To mitigate losses the strategy also close the short position when an up-ward event has been triggered. This imitates a stop loss (it is believed that the market trades up-wards now). The pseudo-code for this strategy is presented in Algorithm 2. From the information presented, it is difficult to earn money when using this strategy (see Question 4).

Algorithm 2 DC_trading(price, Δx_{dc}, Δx_{exit})

$Initialize\ (only\ for\ first\ iteration) : direction = 0,\ x^{ext} = price$
if $direction == 0$ **then** ▷ Only used for starting the algorithm
    if $x^{ext} < price$ **then**
        $x^{ext} = price$
    else
        $diff = (price - x^{ext})/x^{ext}$
        if $diff \leq -\Delta x_{dc}$ **then**
            $direction = -1$
            $x^{ext} = price$
            Open short position
            $short\_price = price$
else if $direction == 1$ **then**
    if $x^{ext} < price$ **then**
        $x^{ext} = price$
    else
        $diff = (price - x^{ext})/x^{ext}$
        if $diff \leq -\Delta x_{dc}$ **then**
            $direction = -1$
            $x^{ext} = price$
            Open short position
            $short\_price = price$
else if $direction == -1$ **then**
    if $x^{ext} > price$ **then**
        $x^{ext} = price$
    else
        $diff = (price - x^{ext})/x^{ext}$
        $close\_diff = (short\_price - price)/short\_price$
        if $diff \geq \Delta x_{dc}$ **then**
            $direction = 1$
            $x^{ext} = price$
            Close short position
        else if $close\_diff \geq \Delta x_{exit}$ **then**
            Close short position

Question 4

In this activity, you test a couple of trading strategies of your choice. You may use any intraday frequency. You should explain how the trading strategies use the underlying statistical properties of the financial data to generate buy-sell signals on tick data. The designed trading strategies will be deployed as algorithmic trading strategies for paper trading using a selected trading platform. Submit a short report on this activity.

Optimizing and back-testing different HFT strategies

All strategies presented has been optimized in cTrader using the genetic algorithm provided. The genetic algorithm was set to maximize net profit and the amount of winning trades while minimizing the max equity drawdown. After optimization the parameters have been used in a back-test on a hold-out test data-set for performance evaluation. All back-tests were conducted on a demo account with a balance of 10,000 € or $.

Directional change strategy

The strategy shown in Algorithm 2 is implemented in cTrader. The strategy is optimized on Bitcoin (BTC/USD) tick-data from 01/01/2023 to 06/06/2023. The best performing strategy is selected and back-tested on Bitcoin tick-data from 06/06/2023 to 06/08/2023. The volume of the strategies are exactly 1 Bitcoin per trade.
The strategy was tested with $\Delta x_{exit} = \infty$. This makes the strategy close the short position only when a long-event occurs. This can be displayed in the following figure (a). Red arrow = open short position and green arrow = close short position:

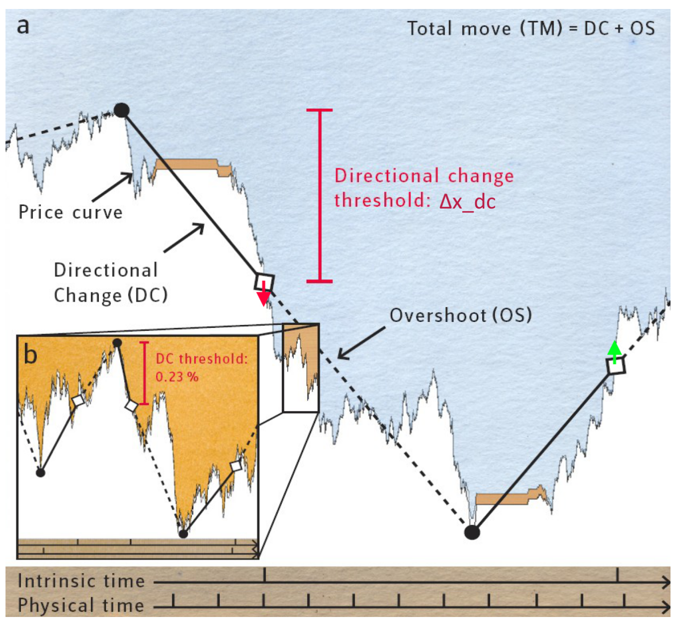

Figure 19: Directional change strategy

Using the scaling-law presented in Question 3 it can be examined when and if this strategy can earn money. To get a rough estimate assume that $\Delta x^{OS} > \Delta x_{dc}$ before this strategy earns money. Lets define the excess return as the difference (here the numbers are taken from Equation OS_short_values):

$$r_e = \left(\frac{1}{C_{x,OS}}\right)^{E_{x,OS}} \cdot \Delta x_{dc}^{E_{x,OS}} - \Delta x_{dc} = 0.587 \cdot \Delta x_{dc}^{0.889} - \Delta x_{dc}$$

The Δx_{dc}'s that produce a positive excess return $r_e \geq 0$ can now be found:

$$0.587 \cdot \Delta x_{dc}^{0.889} \geq \Delta x_{dc}$$
$$\Updownarrow$$
$$0.587 \geq \Delta x_{dc}^{0.111}$$
$$\Updownarrow$$
$$0.587^{9.009} \geq \Delta x_{dc}$$
$$\Updownarrow$$
$$\Delta x_{dc} \leq 0.008$$

Theoretically the strategy makes money when $0 \leq \Delta x_{dc} \leq 0.008$. The maximum excess return is found by differentiation:

$$r_e' = \frac{\partial}{\partial x}(0.587 \cdot x^{0.889} - x) = \frac{0.552}{x^{0.111}} - 1 = 0$$

This is solved by $\Delta x_{dc} = 0.0028$, this will produce an excess return of:

$$r_e = (\frac{0.0028}{1.820})^{0.889} - 0.0028 = 0.00036$$

The maximum excess return for this strategy is therefore found to be 0.036 %. This is easily eaten by the bid-ask spread and the commissions. The strategy needs to be improved to generate more excess return.
From the optimization in cTrader it was found that the best parameters for this model is $\Delta x_{dc} = 0.001$. Which is the lowest value Δx_{dc} was optimized over. From the analysis even lower values might create more return. The back-test yields these results:

Statistic	All trades	Long trades	Short trades
Trades	5255	-	5255
Hit ratio	38.6 %	-	38.6 %
Max equity drawdown	13.8 %	-	-
Sharpe ratio	0.04	-	0.04
Net profit (balance)	7554.52	-	-

Table 1: Results of the directional change strategy on Bitcoin (with $\Delta x_{dc} = 0.001$ and $\Delta x_{exit} = \infty$)

This strategy has no open position at the end of trading, so the net profit (equity) is 7554.52. The strategy creates a lot of trades (5255 for 2 months) and makes a good amount of profit. The Sharpe-ratio is 0.04 which indicates that the strategy takes a lot of risk. Another variation of this strategy was tested. The focus is on profit lost, because a up-trend has to be detected, before the position can be closed. Therefore the parameter Δx_{exit} is used to close the position (hopefully right before an up-trend occurs). An illustration of the strategy can be seen in image a:

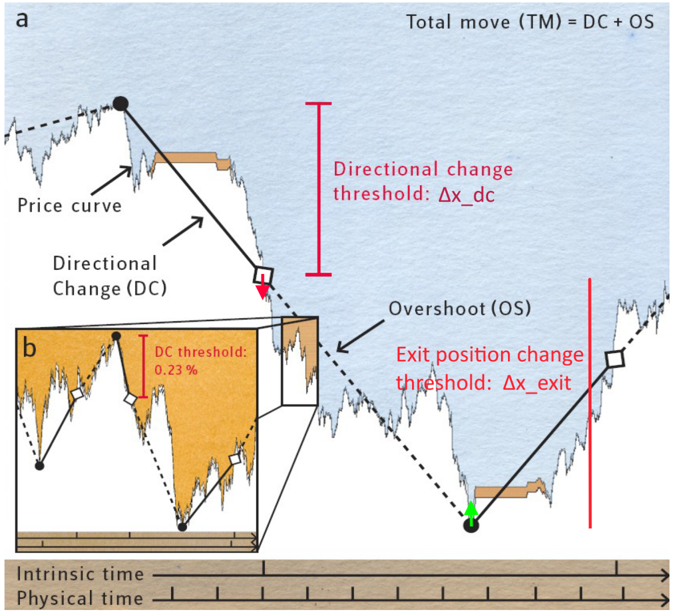

Figure 20: Directional change strategy with Δx_{exit}

From the optimization it was found that the best parameters are $\Delta x_{dc} = 0.001 = 0.1\%$ and $\Delta x_{exit} = 0.016 = 1.6\%$. From the scaling law analysis it is found that $\langle OSV \rangle_1 \approx \Delta x_{dc}$ [16]. It is very unlikely that a price drop of 1.6 % is observed after a directional change of 0.1 %. Therefore this strategy with these parameters is almost identical to the one presented above.

Statistic	All trades	Long trades	Short trades
Trades	5255	-	5255
Hit ratio	38.6 %	-	38.6 %
Max equity drawdown	13.8 %	-	-
Sharpe ratio	0.04	-	0.04
Net profit (balance)	7398.40	-	-

Table 2: Results of the directional change strategy on Bitcoin (with $\Delta x_{dc} = 0.001$ and $\Delta x_{exit} = 0.016$)

This strategy has no open positions at the end of trading, so the net profit (equity) is 7398.40. Since this strategy is similar to the one presented earlier, almost identical results are observed.

Paper-trading with directional change strategy

It was found that the directional change strategy with $\Delta x_{dc} = 0.001$ and $\Delta x_{exit} = \infty$ performs quite well on the optimization and back-test. This strategy was set ”live” into a paper-trading environment. The strategy was tested using Bitcoin tick-data for 16 hours on: 9/8/23. The strategy generated 47 trades, with a hit-ratio of 43 %. The strategy had a profit of 152.94 €, which amount to an average of 3.25 € per trade. From the analysis in cTrader, a commission of 0 € is noted. In the live market the strategy should pay more commission. Additionally there might not be a whole Bitcoin available for buying/selling at the top of book. This will also effect the price if the strategy was used in live trading. Therefore this strategy might not be profitable under live-market conditions.

RSI strategy

RSI is an indicator that identifies if an asset is overbought or oversold. The RSI indicator lies between 0 and 100. If the RSI value crosses an upper-limit: all long positions are close and a short position is opened. If the RSI reaches below a lower-threshold: all short positions are closed and a long position is opened. For more information about the RSI-indicator please refer Jason Fernando (2023) [17].

The strategy was optimized on EUR/USD for different parameters and time-periods. The following parameters were found to be optimal: 3-minute data, RSI minimum: 21, RSI maximum: 74 and a window size of 30 periods. These estimates were used in a back-test which resulted in:

Statistic	All trades	Long trades	Short trades
Trades	2	1	1
Hit ratio	100 %	100 %	100 %
Max equity drawdown	9.99 %	-	-
Sharpe ratio	0.77	-	-
Net profit (balance)	2488.55	-	-

Table 3: Results of the RSI-strategy on EUR/USD for the period: 08/07/2023 - 08/08/2023

This strategy has no open positions at the end of trading, so the net profit (equity) is 2488.55. The Sharpe-ratio was acceptable at 0.77. This strategy conducted two trades only. Therefore it can't be concluded that the strategy would perform well in live-trading.

Moving average (MA) strategy

This strategy uses a fast (few periods) and slow (many periods) moving average to generate signals. If the fast MA crosses the slow MA from below, a buy signal is created (go long, close short). Contrary if the fast MA crosses the slow MA from above, then a sell signal is created (go short, close long). The strategy was optimized on EUR/USD. The following parameters were obtained: 8-minute data, fast MA periods: 16, slow MA periods: 70, stop loss: 16, take profit: 4, max orders: 1 and lot-value: 2. These estimates were used in a back-test on a hold-out data-set:

Statistic	All trades	Long trades	Short trades
Trades	73	36	37
Hit ratio	84.9 %	83.3 %	86.5 %
Max equity drawdown	8.84 %	-	-
Sharpe ratio	0.12	-	-
Net profit (balance)	1270.62	-	-

Table 4: Results of the MA-strategy on EUR/USD for the period: 08/06/2023 - 08/07/2023

This strategy has no open positions at the end of trading, so the net profit (equity) is 1270.62. This strategy has a good amount of trades for a month worth of back-testing and with a decent hit-ratio. Though the Sharpe-ratio is low. Therefore it is difficult to conclude if this strategy is going to perform well in live market conditions.

Bollinger bands (BB) strategy

Bollinger bands can be created using a MA, an upper and lower band. These bands can be found by $MA + m \cdot SD$, where SD is the volatility for a period and "m" is a parameter. If the price gets outside the bands, then a buy or sell signal is created. For more information on Bollinger bands please refer to Adam Hayes (2023) [18]. The parameters found from the optimization process (on EUR/USD): Band periods: 12, standard deviations: 2.2, MA type: weighted, initial volume percent: 0.7, stop loss: 105 and take profit: 100. These estimates were used in a back-test on a hold-out data-set:

Statistic	All trades	Long trades	Short trades
Trades	11	6	5
Hit ratio	63.6 %	66.6 %	60.0 %
Max equity drawdown	11.74	-	-
Sharpe ratio	0.18	-	-
Net profit (balance)	1400.99	-	-

Table 5: Results of the BB-strategy on EUR/USD for the period: 02/04/2023 - 01/05/2023

This strategy has some open positions at the end of trading, so the net profit (equity) is 1476.64. With the low Sharpe-ratio and few trades, it is difficult to conclude, if the strategy would continue earning profits in the markets.

Question 5

Design and deploy a trading strategy based on certain characteristics of high frequency data (intraday data of your choice). You should explain how the trading strategy uses the underlying statistical properties of the financial data to generate buy-sell signals on tick data. Submit a very short report on this activity and the backtesting results. Note that third party trading strategies are not allowed. That means you should write your own code (any programming language of your choice) and do the back testing on the selected data.

The trading strategy introduced in Question 4: Directional change strategy is expanded. The first addition to this strategy is to open a long-position when an up-trend has been detected. Additionally the strategy is improved with a larger scale understanding. The idea can be explained using this figure:

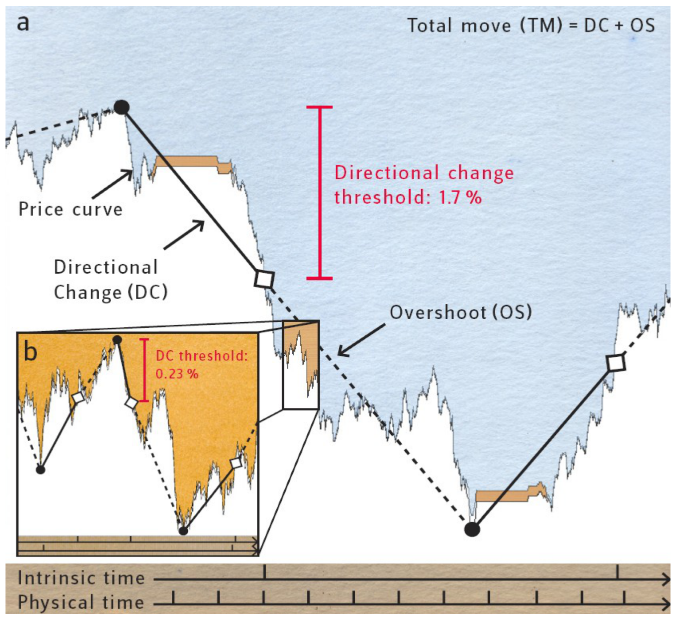

Figure 21: Display of directional changes, taken from: [16]

Assume that the strategy trade based on the scale presented in image b, then it might be useful to know what direction is active on large scale as seen in image a. An improvement proposal is to enter a short position when a down-ward trend is detected on small scale and during a down-ward trend on large scale. In the same way only enter a long position if both directions on small and big scales are up-wards. The trend on the large scale is found by the direction change using the threshold: Δx_{dc}^{big}. Another new idea is to consider the last n (eg. n = 25) events. Of these n events, there will be "w" winning and "l" losing trades. It is assumed that the winning trades happen with probability p. Therefore it can be assumed that: $w|p \sim binomial(n, p)$. In Bayesian statistics a conjugate prior to this likelihood is the beta distribution [19, page 42-43]. This yield the posterior:

$$p|w \sim beta(w + a, n - w + b)$$

Here a and b are prior parameters. The mean of this beta distribution can be found:

$$\hat{p} = E(p|w) = \frac{w + a}{n + a + b}$$

Using the overshoot value data, this probability can be estimated using a rolling window over the events (see code in this Github-link):

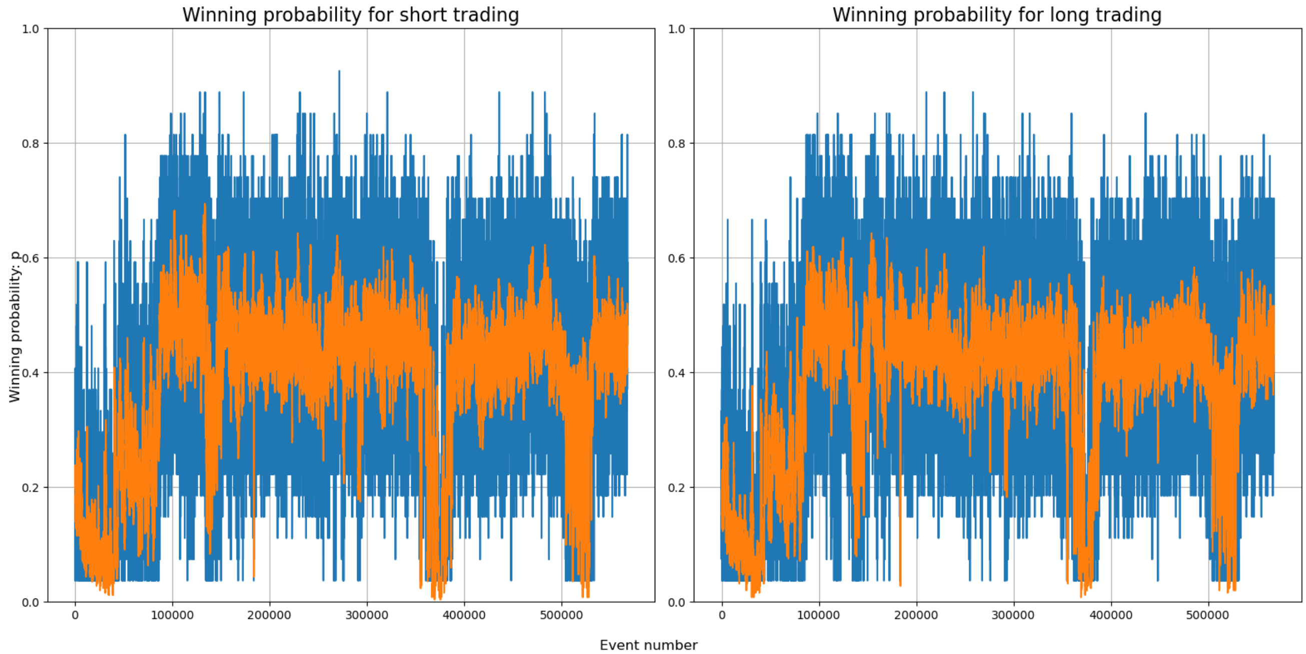

Figure 22: Winning probability for $\Delta x_{dc} = 0.001$

The blue line illustrates the winning probability over a period of n = 25, and the period for the orange line is 250. For these plots an uninformed prior has been set (a = b = 1). These plots illustrate that the winning probability changes over time. This indicates that the strategy enters "winning" and "losing" streaks. The idea is to start trading when a winning streak occur and stop trading throughout the time of a losing streak. To achieve this the winning probability should be above some threshold (p_win) before trading. The new additions of information creates the new parameters: large scale DC threshold (Δx_{dc}^{big}), rolling window size (n), a threshold for the winning probability (p_win) and the prior parameters: a and b.

The pseudo-code for the trading strategy can be written:

Algorithm 3 DC_prob(price, Δx_{dc}, Δx_{dc}^{big}, n, p_win, $a = 1$, $b = 1$)

```
Initialize (only for first iteration) : direction = 0,  x^ext = price
p̂_long = (w_long + a)/(n + a + b)
p̂_short = (w_short + a)/(n + a + b)
direction_big = DirectionChangeFunction(Δx_dc^big)
if direction == 0 then                                  ▷ Only used for starting the algorithm
    if x^ext < price then
        x^ext = price
    else
        diff = (price − x^ext)/x^ext
        if diff ≤ −Δx_dc then
            direction = −1
            x^ext = price
            if p_win < p̂_short and direction_big == −1 then
                Open short position
else if direction == 1 then
    if x^ext < price then
        x^ext = price
    else
        diff = (price − x^ext)/x^ext
        if diff ≤ −Δx_dc then
            direction = −1
            x^ext = price
            if p_win < p̂_short and direction_big == −1 then
                Open short position
            Close long position
else if direction == −1 then
    if x^ext > price then
        x^ext = price
    else
        diff = (price − x^ext)/x^ext
        if diff ≥ Δx_dc then
            direction = 1
            x^ext = price
            if p_win < p̂_long and direction_big == 1 then
                Open long position
            Close short position
```

It should be noted that the rolling window size (n) is less than specified for the first n-events. For each event (up- or down-trend) the strategy monitors if a position would generate a win or loss. This is even if no position is taken (if e.g. $p\_win > \hat{p}\_long$). In such scenarios the event-prices are noted to calculate profit/loss. This strategy is written in C#. See the code in this Github-link or the "dc_trader_bayes" file in material. The strategy is optimized in cTrader on Bitcoin (BTC/USD) tick-data from 01/01/2023 to 06/06/2023. Here the best parameters are found to be: $\Delta x_{dc} = 0.001$, $\Delta x_{dc}^{big} = 0.001$, $n = 51$ and $p\_win = 0.3$. The prior used was uninformed because the parameters was hard-coded to one ($a = 1$ and $b = 1$).

With the optimal parameters the strategy was back-tested on the hold-out set (06/06/2023 - 06/08/2023):

Statistic	All trades	Long trades	Short trades
Trades	10246	5067	5179
Hit ratio	38.7 %	38.8 %	38.6 %
Max equity drawdown	12.76 %	-	-
Sharpe ratio	0.05	0.06	0.04
Net profit (balance)	17336.48	9862.39	7474.09

Table 6: Results of the strategy on Bitcoin (with $\Delta x_{dc} = 0.001$, $\Delta x_{dc}^{big} = 0.001$, $n = 51$, $p\_win = 0.3$ and $a = b = 1$)

This strategy has no open positions at the end of trading, so the net profit (equity) is 17336.48. The introduction of long-trading netted the strategy nearly ten thousand in profits. The Sharpe-ratio was 0.05 which is somewhat low. Therefore it can't be concluded that the strategy would perform well in live-trading. The DC threshold parameters are equal: $\Delta x_{dc} = \Delta x_{dc}^{big} = 0.001$. This means that the strategy does not reject trades using the "larger-scale" information. In the optimization it was found that reducing the window size (n) from 101 to 51 (with $p\_win = 0.3$) would result in 1148 fewer trades with 772 of them being losing trades. These 1148 trades would have cost -2367.41 in profits and therefore n is 51 in the optimal strategy. This indicates that the Bayesian setup helps reducing trades during "losing" streaks.

Paper-trading with the directional change strategies

The trading strategy with the parameters $\Delta x_{dc} = 0.001$, $\Delta x_{dc}^{big} = 0.003$, $n = 51$ and $p\_win = 0.1$ performed decent in the optimization. This strategy uses the "large-scale" information of $\Delta x_{dc}^{big} = 0.003$. This strategy will be referred to as the "large strategy".

The new optimal strategy, the large strategy and the old directional change strategy was set "live" in a paper-trading environment from 15/8/2023 to 18/8/2023 on Bitcoin tick-data.

The old strategy generated 428 trades, with a hit-ratio of 26.2 %. The strategy had a profit-loss of 11016.41 €.
The large strategy generated 357 trades, with a hit-ratio of 27.5 %. The strategy had a profit-loss of 8914.96 €.
The optimal strategy generated 429 trades, with a hit-ratio of 21.9 %. The strategy had a profit-loss of 13655.52 €.
From the results it is clear that the strategies enter many bad trades which result in large losses. It can therefore be seen that even if a trading strategy performs well in back-testing no guarantee is given of replication in live-markets. This illustrates the importance of paper-trading as a quality measure. Other factors (e.g. more commission, etc.) might also reduce the performance in live-market conditions.

If the strategies tested had better performance (e.g. better Sharpe-ratio), then the Kelly criterion for position-sizing is also an improvement suggestion [20].

References

[1] Chen James. Understanding the lemons problem and how to solve it. *Investopedia*, 2021. Accessed: 2023-31-7. URL: https://www.investopedia.com/terms/l/lemons-problem.asp.

[2] Chinthalapati Raju. Market micro-structure theory: information models (slides: Week1d), 2023.

[3] The Investopedia Team. Market failure: What it is in economics, common types, and causes. *Investopedia*, 2023. Accessed: 2023-31-7. URL: https://www.investopedia.com/terms/m/marketfailure.asp.

[4] Palmer Barclay. What determines a stock's bid-ask spread? *Investopedia*, 2023. Accessed: 2023-31-7. URL: https://www.investopedia.com/ask/answers/06/bidaskspread.asp.

[5] Glosten Lawrence R. and Milgrom Paul R. Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of financial economics*, 14(1):71–100, 1985. URL: https://soeg.kb.dk/permalink/45KBDK_KGL/1f0go08/cdi_crossref_primary_10_1016_0304_405X_85_90044_3.

[6] Chinthalapati Raju. High frequency finance & algorithmic trading (slides: Week1a), 2023.

[7] Aldridge Irene. *High-frequency trading: a practical guide to algorithmic strategies and trading systems, second edition*. Wiley trading series. Wiley, Hoboken, N.J, 2nd ed. edition, 2013. URL: https://soeg.kb.dk/permalink/45KBDK_KGL/1f0go08/cdi_skillsoft_books24x7_bkf00052749.

[8] Hansen Peter R and Lunde Asger. Realized variance and market microstructure noise. *Journal of business & economic statistics*, 24(2):127–161, 2006. URL: https://soeg.kb.dk/permalink/45KBDK_KGL/1f0go08/cdi_scopus_primary_43706495.

[9] Ait-Sahalia Yacine. *High Frequency Market Microstructure Noise Estimates and Liquidity Measures*. NBER working paper series no. w13825. National Bureau of Economic Research, Cambridge, Mass, 2008. URL: https://soeg.kb.dk/permalink/45KBDK_KGL/1pioq0f/alma99123043235605763.

[10] Chinthalapati Raju. Quantitative methods for high frequency data (slides: Week2b), 2023.

[11] Zucchini Walter, MacDonald Iain L, and Langrock Roland. *Hidden Markov Models for Time Series: An Introduction Using R, Second Edition*, volume 150 of *Chapman & Hall/CRC Monographs on Statistics and Applied Probability*. CRC Press, Oakville, 2017. URL: https://soeg.kb.dk/permalink/45KBDK_KGL/1f0go08/cdi_informaworld_taylorfrancisbooks_b20790.

[12] Lewinson Eryk. *Python for Finance Cookbook*. Packt Publishing, Birmingham, England, 2020. URL: https://soeg.kb.dk/permalink/45KBDK_KGL/1f0go08/cdi_safari_books_v2_9781789618518.

[13] Wikipedia contributors. Scale invariance — Wikipedia, the free encyclopedia, 2023. Accessed: 2023-08-03. URL: https://en.wikipedia.org/w/index.php?title=Scale_invariance&oldid=1160187039.

[14] Bouchaud Jean-Philippe. Power-laws in economy and finance: some ideas from physics, 2000. arXiv:cond-mat/0008103.

[15] J. B. Glattfelder, A. Dupuis, and R. B. Olsen. Patterns in high-frequency FX data: discovery of 12 empirical scaling laws. *Quantitative Finance*, 11(4):599–614, apr 2011. URL: https://doi.org/10.1080%2F14697688.2010.481632, doi:10.1080/14697688.2010.481632.

[16] Chinthalapati Raju. Directional changes (fractals) for algorithmic trading (slides: Week2c), 2023.

[17] Fernando Jason. Relative strength index (rsi) indicator explained with formula. *Investopedia*, 2023. Accessed: 2023-8-9. URL: https://www.investopedia.com/terms/r/rsi.asp.

[18] Hayes Adam. Bollinger bands®: What they are, and what they tell investors. *Investopedia*, 2023. Accessed: 2023-8-9. URL: https://www.investopedia.com/terms/b/bollingerbands.asp.

[19] Reich Brian J. *Bayesian statistical methods*. Texts in statistical science. CRC Press, Taylor & Francis Group, Boca Raton, 2019 - 2019. URL: https://soeg.kb.dk/permalink/45KBDK_KGL/1pioq0f/alma99124023326505763.

[20] Chinthalapati Raju. The kelly criterion how to manage your money when you have an edge (slides: Week1e), 2023.