Quantitative Finance Strategies and Experience (Dr Michael Nelson)

In this document I aim to briefly summarise the research and strategy development I have worked on as a quantitative researcher at both Winton Capital Management and Voltz Labs, providing brief details and preliminary results. While internal strategy code cannot be shared, I will provide links to any publicly-available code pertaining to my research. Within Voltz labs my initial focus was on the development of the infrastructure for the Voltz automated market maker, the risk analysis and models for this market maker, and the central strategy backtesting framework for future strategy testing (April − June, 2022); the more recent work has focused on research into yield arbitrage, trend-following, and statistical arbitrage strategies and how they could be applied to fixed-income decentralised finance (July, 2022 − present). Following discussion my experience at Voltz Labs, I'll provide an overview of my work at Winton Capital, as well as general thoughts on future trading strategies.

Voltz interest rate swap automated market making strategy

My early research as a quant at Voltz Labs focused on aiding in the development of the Voltz virtual automated market maker (AMM), and corresponding margin engine, based on the concentrated liquidity (AMM) in Uniswap v3 [1]. The Voltz AMM creates an interest rate swaps market by allowing for the exchange of 1 % fixed rate tokens for variable rate tokens based on the particular coin or token of interest. Example: allowing a user to exchange a lending or borrowing APY on Aave USDC for a fixed USDC rate set by the constant product AMM mechanic. At Voltz we have now established IRS pools covering a series of stablecoins and liquid staking ETH derivatives: Aave USDC, Aave DAI and, Compound DAI lending rates; Aave USDC, Compound DAI, and Compound USDT borrowing rates; Lido stETH and Rocket rETH lending rates, where each pool has a unique time to maturity. The Uniswap v3 feaures allow for virtual reserves to be created for each Voltz pool (see Figure 1), where liquidity providers deposit liquidity in tick ranges of rate (rather the price, as per Uniswap), providing concentrated liquidity and improving protocol capital efficiency. In the virtual pool we may use the standard convex pricing result $P = -dy/dx = y/x$ for a fixed-rate x-axis and variable y-axis to compute the implied fixed rate:

$$\text{Implied fixed rate} = \frac{1}{\text{AMM price}}, \tag{1}$$

taking a variable rate movement of 1 unit without loss of generality. With Voltz IRS positions, it is now possible to form fixed-income strategies where one effectively takes a long or shot position in the variable rate, based on some margin M deposited to generate a leveraged position, N/M, for notional amount N. There are three positions in this strategy for v variable tokens, f fixed tokens, and time t in the pool:

- Fixed-taker (FT): deposit variable tokens to lock in and received fixed tokens. Payoff: $N(f - v)t$ i.e. a short position on the variable APY of a token from a lending/borrowing protocol like Aave;
- Variable-taker (VT): deposit fixed tokens to lock in and received variable tokens. Payoff: $N(v - f)t$ i.e. a long position on the variable APY;
- Liquidity provider (LP): deposit notional, converted into fixed and variable tokens, in order to earn 30 basis points liquidity rewards (similar to the older Uniswap rewards scheme for LPs, with 0.30 % transaction costs to shift impermanent loss, before fee bands where introduced in Uniswap v3).

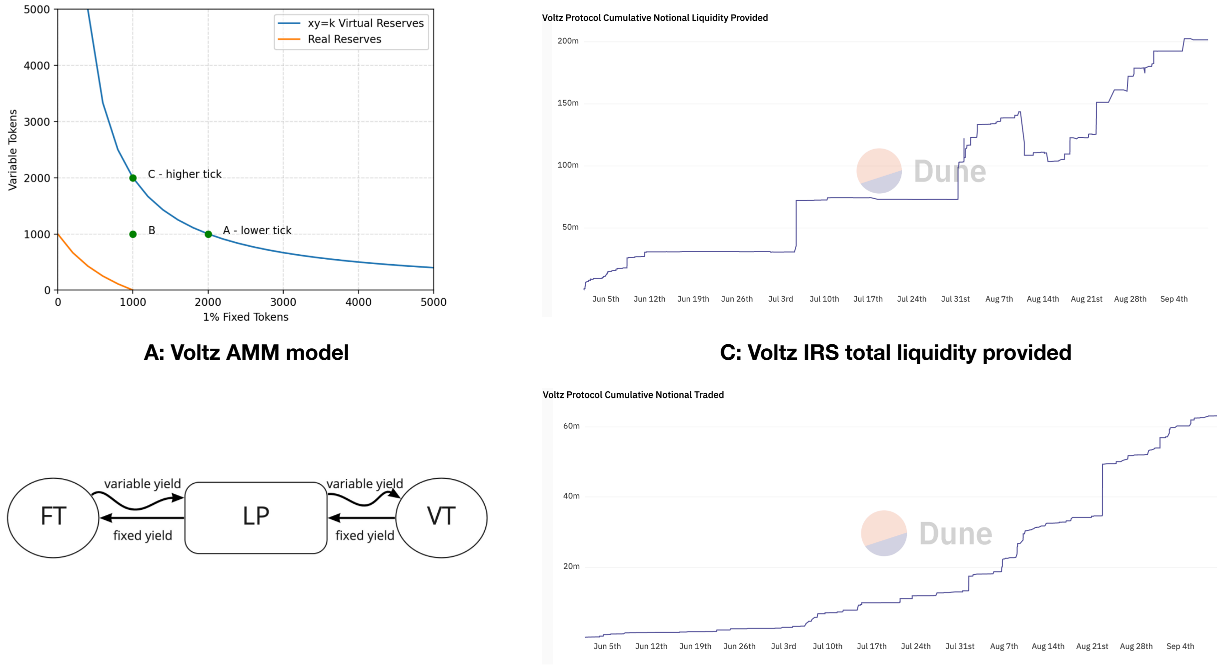
Figure 1 summarises the Voltz AMM, and the total liquidity and notional traded as of early September, 2022 (the first Voltz pools began trading June, 2022).

Statistical arbitrage in DeFi rates trading

My recent focus as a Quant at Voltz has been the development of live trading infrastructure and associated quantitative strategies for proprietary trading. I am leading all research and implementation around these strategies, covering statistical arbitrage, trend-following, and yield arbitrage/carry. Statistical arbitrage strategy development takes two strands: statistical arbitrage signals build on token APY (i.e, variable lending/borrowing rate) ratios; and signals built on token APY spreads.

Statistical arbitrage with ratios

We first form relevant token pairs by checking for the cointegration between historical rates series over a given lookback window (take a 30-day window). The APYs used as historical rates are obtained through a 1-day



A: Voltz AMM model

C: Voltz IRS total liquidity provided

B: Voltz IRS FT-VT-LP mechanism

D: Voltz IRS total notional traded

Figure 1: Summary of the Voltz AMM interest rate swap model and analytics.

lookback window applied to the raw liquidity indices extracted using rate oracles from the lending/borrrowing protocol. We investigated two different lookback implementations of our pairs-trading statistical arbitrage strategy, monthly and rolling. In the monthly signal construction, we build a signal for a given timestamp (e.g. present day, t) using the historical rates date from the previous month and only the previous month, so the final signal value is only updated when we enter a new month in the trade, and the present value of the rates is checked against this fixed signal each day. In the rolling signal construction, in a similar manner to our trend and rates trading (see subsequent sections), we construct the same statistical arbitrage signal but now using the previous N days historical rates data from the present. This signal value is updated daily as we roll from day N to $N+1$. In both monthly and rolling cases, the signal calculation itself is the same. Given a set of historical APYs (e.g. previous month or rolling previous 30 days) we compute the ratio of those APYs, and standardise this quantity into a Z-score:

$$R_t = \frac{\text{APY}_t^{\text{Token 1}}}{\text{APY}_t^{\text{Token 2}}}, \tag{2}$$

and

$$Z_t = \frac{R_t - \mu_R}{\sigma_R}. \tag{3}$$

Here the μ_R and σ_R are the means and standard deviations over the given historical APY series. At current time t we then compare today's ratio with this historical ratio and its Z-score. Trades are made based on when today's ratio of pairs is d standard deviations away from the historical mean Z-score i.e.

$$S^{\pm}_{\text{Ratio}} = \frac{1}{L}\sum_{i=t-L-1}^{t-1} Z_i \pm d\sigma_Z. \tag{4}$$

Here the nature of the L lookback depends on the monthly/rolling strategy, and by default we look for differences between today's Z-score ratio and the historic mean above/below one standard deviation of that mean, i.e. $d = 1$. Figure 2 summarises the position-taking logic for the statistical arbitrage signals, in pseudocode. At each point in trading, we run an Augmented Dickey-Fuller test [2] to check for statistically significant evidence of cointegration between the tokens considered in the pairs trade. Long positions are taken using token lending APYs, with short positions filled using token borrowing APYs.

In Figure 3 we can see that, between 11.2021 and 03.2022, this ratio-based strategy generates returns of 6.46 % with some drawdown of 0.17 % over 11 days.

If (Token1, Token2) cointegrate :

If $Z_t > S^{+}_{\text{stat arb}, t}$: SHORT T1, LONG T2

Else if $Z_t < S^{-}_{\text{stat arb}, t}$: LONG T1, SHORT T2

Else : EXIT position

Else : EXIT position

Figure 2: Pseudocode of the long/short execution plan for the ratio strategy on positions at any given timestamp, t. Positions are unwound when crossing a Z-score of zero from either direction of the mean-reverting cycle.

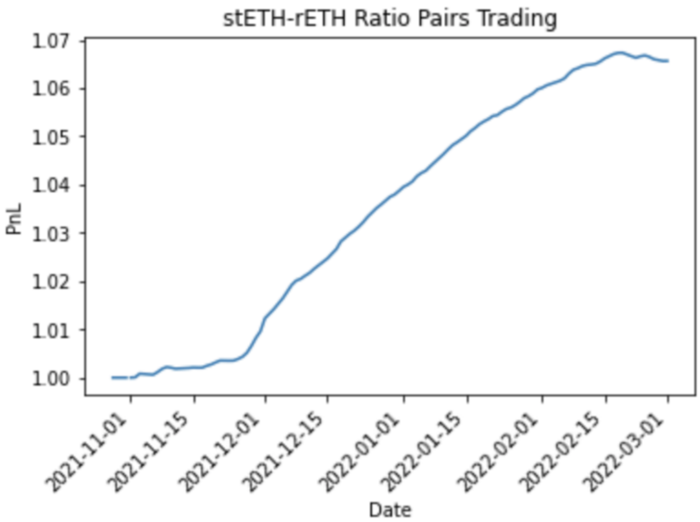


Figure 3: The profit and loss curve in a statistical arbitrage ratio trade for the stETH-rETH token pair.

The major disadvantage is that we only ever take a single given long position and short position at any given time. Although this is position-neutral, the overall portfolio isn't necessarily hedged correctly i.e. perhaps we need to take >1 short positions to be hedged more appropriately. The most natural way to incorporate a hedge is by considering the spread between assets.

Statistical arbitrage with spreads

In the previous strategy, I developed a Z-score from the ratio of rates, based on the idea that it is approximately mean-reverting. The same could be done based on the spread, s, between two rates:

$$s = \text{APY}_t^{\text{Token 1}} - \gamma \text{APY}_t^{\text{Token 2}} \tag{5}$$

where we have now added an additional parameter, γ, which is the amount of token 2 we need to short such that the overall spread is mean-reverting; this is our hedging factor. There are different ways to compute the hedging factor, using a variety of statistical models. In the simplest case, we may compute some hedging factor between two asset rates from their historical time series using a linear regression model. There is however a problem here: such a strategy assumes a constant hedging factor but we expect market conditions on the rates to change, generating changing drifts and volatilities on the underlying APYs, and therefore changing the degree of hedging in accordance with different bull or bear dynamics which emerge over time. We can therefore extend this static hedging approach and build a model to dynamically update $\gamma \rightarrow \gamma_t$ such that the spread remains mean-reverting as the hedging factor evolves. I am currently developing and implementing strategies to trade lending and borrow rates of cointegrated pairs with Kalman filter [3] and Johansen test [4] updates of the hedge.

The Kalman filter is a linear state-space model, which computes the values of "hidden" state variables of the system over time. Here the state variables are the hedging factors of different assets in a portfolio. We want to be able to update the hedging factors each day as a new piece of rate data arrives from the market/blockchain. Since we can build a spread from many assets, lets define the state vector as the vector of all M assets' hedging factors. This hidden state vector, θ_t, is updated according to the application of a transition matrix G_t:

$$\theta_t = G_t^T \theta_{t-1} + w_t, \tag{6}$$

adding a w_t term to model the Gaussian noise in our system.

The model learns through updating the observation when the hidden state is updated with a new token APY received from the relevant lending or borrowing protocol. The update for this observation y_t is also assumed to be linear:

$$y_t = F_t^T \theta_t + v_t \tag{7}$$

adding a v_t measurement error and an F_t observation matrix.

The model updates are based on the covariances of the parameters; full computational details are provided in the Voltz internal strategies classes, but I am not permitted to share these details at this point in time, as strategies are under active development.

The Kalman filter assumes the underlying parameters of the model have Gaussian priors, and the entire filter can be thought of as a Bayesian update [5] of the state variable model parameters at time zero, together with Gaussian models of the system noise w_t and measurement error v_t :

$$\theta_0 \sim \mathcal{N}(m_0, c_0), \ v_t \sim \mathcal{N}(0, c_v), \ w_t \sim \mathcal{N}(0, c_w), \tag{8}$$

where the c denotes different covariance matrices between the M assets in the portfolio, and m_0 is some initialisation vector of the hedging factors for the assets.

Once the Kalman filter prediction for the observed true rate of y_t (which we denote Y_t) is computed, the trading signal is then calculated from the Z-score of the spread $s_t = y_t - Y_t$. The mean and standard deviation in the spread are computed from the moving average and moving standard deviation of the past L days of results:

$$Z_{s_t} = \frac{s_t - \mu^s_{t-L+1:t}}{\sigma^s_{t-L+1:t}}, \tag{9}$$

where the $\mu^s_{t-L+1:t}$ and $\sigma^s_{t-L+1:t}$ correspond to the L-day moving averages and standard deviations for the spread.

Using Z_{s_t} we may trade according to the standard statistical arbitrage trading rules, adapting a version of the rules we used for the simpler rate ratios arbitrage trade. Put concretely:

- $Z_{s_t} < -d$: long spread; exit long when the score moves above 0;
- $Z_{s_t} > d$: short spread; exit short when the score moves below 0, where d is the critical number of standard deviations for significance.

By using dynamic hedging, it is now possible to compute mean-reverting portfolios from stETH-rETH and aUSDC-aDAI pairs, which in the static hedging approach was proving difficult. In a backtest from 10.2021 to 07.2022, we generate signals based on the Z-score of the cointegrated spread to trade on, which clearly mean-revert over the time period of several days (a lookback window of 5 days is used in the moving calculations of the scores, but slightly larger lookback windows yield similar results as a checked of parameter robustness). These mean-reverting signals are provided in Figure 4.

A separate dynamic hedging strategy in the works applies the Johansen test to a matrix of rates for M assets across an L lookback period. The test identifies the components required to form a cointegrating portfolio by performing an eigenvalue decomposition on the rates with a vector error correction model (VECM) [6]. Assuming a positive-definite matrix is defined, the leading terms in the M eigenvectors correspond to the

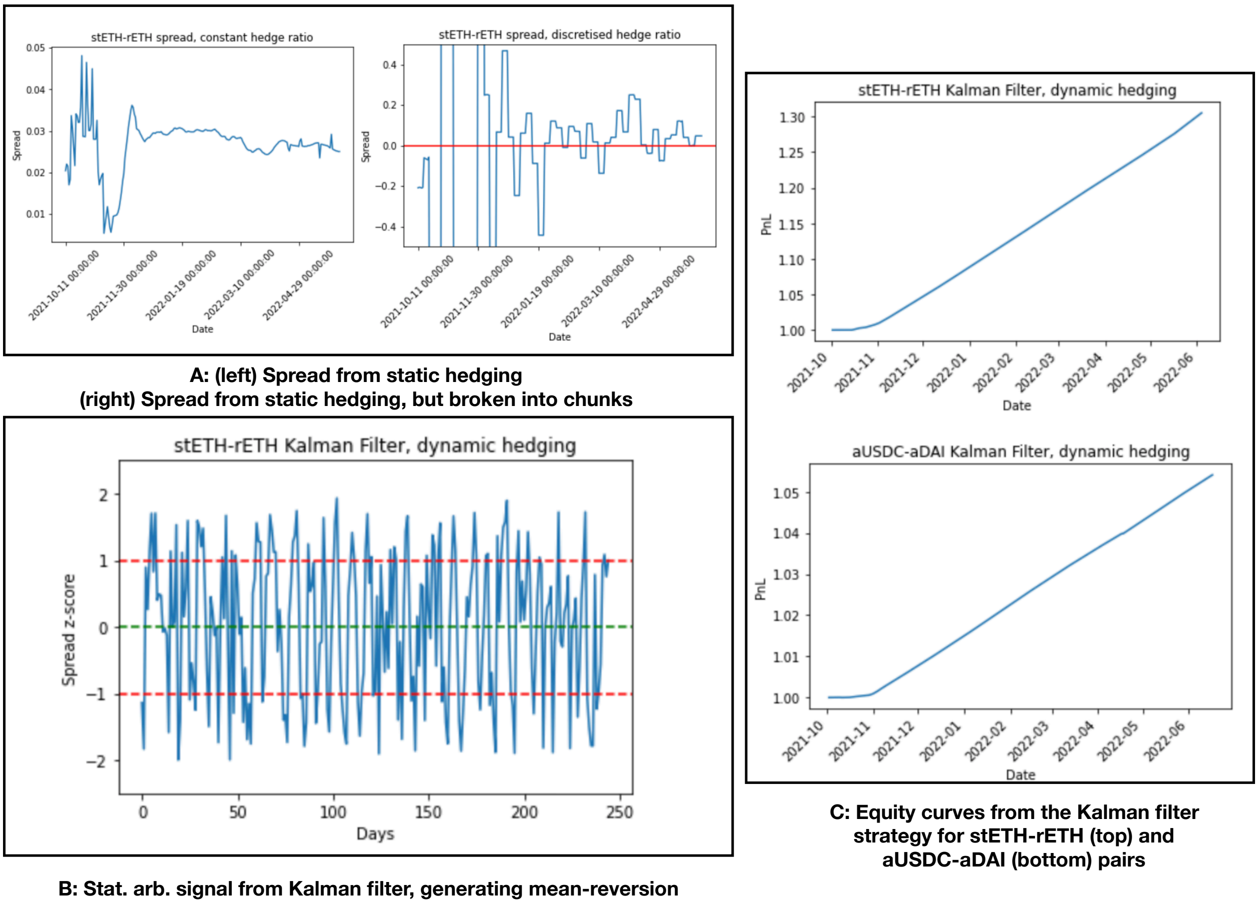
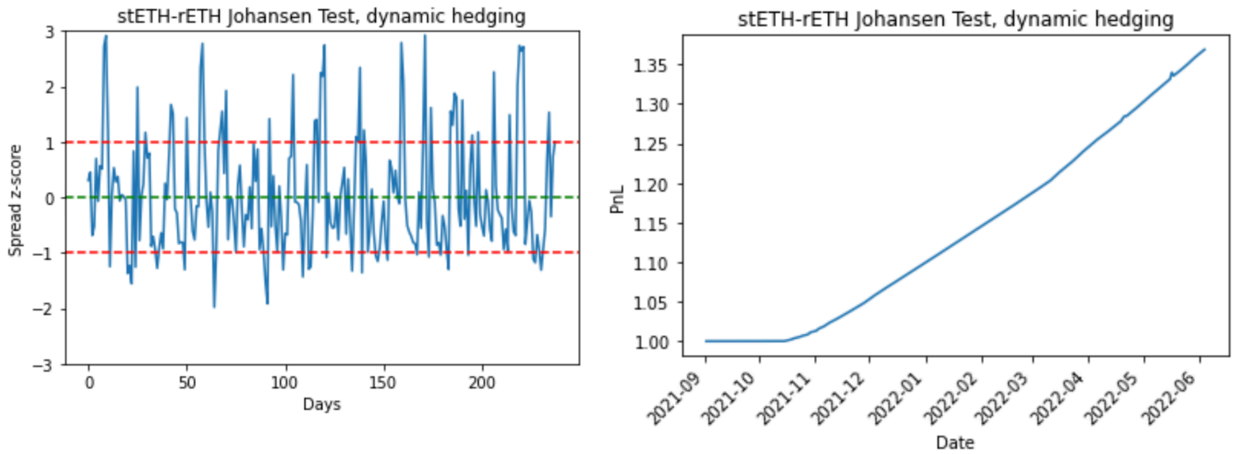


Figure 4: The dynamically hedged spreads for stETH-rETH and aUSDC-aDAI pairs using a Kalman filter to update the hedging factor, and associated equity curves. A comparison of the “mean-reversion" from static hedging is included, indicating the loss of mean-reversion with that approach.



A: Spread Z-score **B: Equity curve**

Figure 5: The dynamically-hedged spread and returns curve for the stETH-rETH pair following statistical arbitrage with position sizing determined from the leading eigenvectors extracted with a Johansen test.

positions necessary to form the mean-reverting portfolio and their associated hedging factors. The signal spread can then be computed:

$$s_t = \sum_i e_i^t[l]\mathrm{APY}_t^i \tag{10}$$

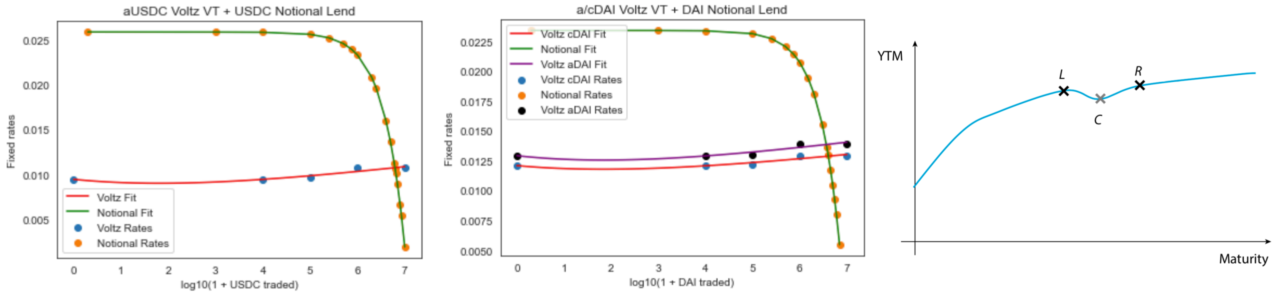
where $e_i^t[l]$ denotes the leading eigenvector component for token i at time t, which has a corresponding rate APY_t^i. Again the Z-score on this spread is used to compute the over- and under-valued trading signal in the statistical arbitrage trade. For the stETH-rETH pair, this signal is mean-reverting over the 10.2021 to 06.2022 backtest period, covering both bull and bear market scenarios, as shown in Figure 5.

For future developments of this strategy, it would be interesting to investigate the application of CNNs and Transformer architectures to statistical arbitrage DeFi portfolios. This is inspired by [7], where a CNN is used to learn short-term trends and reversions in the equity time series and the Transformer learns longer-term temporal trends as "attention heads" between the latent components learned in the CNN filter.

Yield arbitrage with interest rate swaps and DeFi fixed rates

Another strategy under development takes advantages of several DeFi fixed-income protocols, many of which are building products to trade which act like zero-coupon bonds across different maturities, and with different stablecoins and staked ETH underlyings. Two interesting protocols are Element and Notional Finance. Notional Finance acts like a zero-coupon bond protocol, where as Element uses an approach called yield-stripping to separate a yield-bearing token into a variable rate "yield token" (YT) and fixed-rate "principal token" (PT). This is similar in behaviour to strips in traditional finance. The PT can also be treated like a zero-coupon bond in this case, so both the Element PT and the Notional bond objects will be referred to as ZCBs for brevity.

IRS portfolios generated through Voltz can be combined with the bonds from protocols such as Notional and Element Finance. The central idea underpinning this strategy is a yield arbitrage trade, where first a yield curve may be bootstrapped or interpolated (e.g. using the Nelson-Siegel approach [8] and related methods) for a particular bond or PT across different times to maturity, where each bond will correspond to some lending or borrowing rate on the underlying token or coin. Using this fixed lending or fixed borrowing yield curve, we may then compare positions in the Voltz fixed rate market yield *point* with the corresponding maturity on the yield curve, identifying points of vertical arbitrage where the difference between the Voltz fixed rate and the PT/ZCB rate, for the same underlying, is non-zero. This vertical trade would correspond to something like the classic butterfly trade in fixed-income arbitrage, used to close vertical gaps between points of similar maturities on the yield curve. In a similar spirit, it is also possible to perform a carry trade using the same type of PT/ZCB from the fixed-income DeFi protocol, but at different maturities, essentially betting on he future direction and



A: USDC slippage (Voltz aUSDC pool)

B: DAI slippage (Voltz aDAI and Voltz cDAI pools)

C: Yield arbitrage butterfly trade, from Pedersen

Figure 6: The slippage curves for the fixed rate when making a VT trade on Voltz, and lending on Notional Finance, for USDC and DAI stablecoins. A schematic of the butterfly trade is also provided as a comparison, taken from Pedersen [9].

convexity of the DeFi yield curve. One important consideration in the vertical arbitrage strategy is the impact of taking large trading orders on the bond and IRS fixed-rates bought/sold in the trade i.e. slippage. If there is a vertical gap in the yield curve between the Voltz IRS yield point and the ZCB, a large enough trade will generate enough slippage in each rate direction to effectively close the yield gap. To my knowledge, DeFi yield curves are not being actively calculated, presenting an opportunity to get in early.

To provide a more detailed example, consider taking a position as a variable rate taker on the Voltz AMM for aUSDC, aDAI, and cDAI stablecoin IRS pools, and lending at a fixed APR for USDC and DAI on Notional Finance, effectively selling a ZCB. The slippage behaviour associated with this trade can be modelled by extracting on-chain rates data from Voltz and Notional, and fitting a linear model in log space: slippage $= \alpha + \beta \log(1+N)$ for N notional traded in each protocol. The estimated slippages are shown for the aUSDC, aDAI, and cDAI pools in Figure 6, calculated from data scraped from Ethereum. There is an identifiable trade which can close he gap between the lending rate for each token on Notional and the corresponding fixed rate on Voltz, if the trader is prepared o enter into a Voltz VT. This amount this corresponds to about 6.25M USDC for the aUSDC-USDC trade, and about 4.0M of notional DAI for the a/cDAI-DAI trades.

Another application can be to use the position in Notional or Element as a hedge against downside due to interest rate risk which a trader is exposed to when participating in a Voltz IRS. As an example, in Voltz one can can take a VT or FT position on the rates of Lido's stETH. An stETH VT has a linear payoff function dropping below zero when the fixed rate (which the VT holder pays, and which is calculated through the Voltz AMM mechanic) is higher than the variable lending rate of stETH received, as calculated by Lido and used in a rate oracle on Voltz. By buying an stETH PT on Element Finance in parallel to taking the stETH VT position on Voltz, is is possible to generate a quadratic payoff function which outperforms the pure Voltz VT when the VT component is in-the-money, and has a limited downside when the VT component is out-of-the-money. The projected APY for such a portfolio, as a function of the average variable stETH rate, is shown in Figure 7.

DeFi rates trading with trend-following

A preliminary strategy in the works focuses on using the historical size of the lending/borrowing rate of a particular token, and the gradient of that rate over a lookback window, to take long and short position on the variable APY of that protocol token. Such as approach is inspired by trend-following approaches ubiquitous to traditional finance. There are two forms of signals under investigation and optimisation, referred to as S_{trend} and S_{rate}.

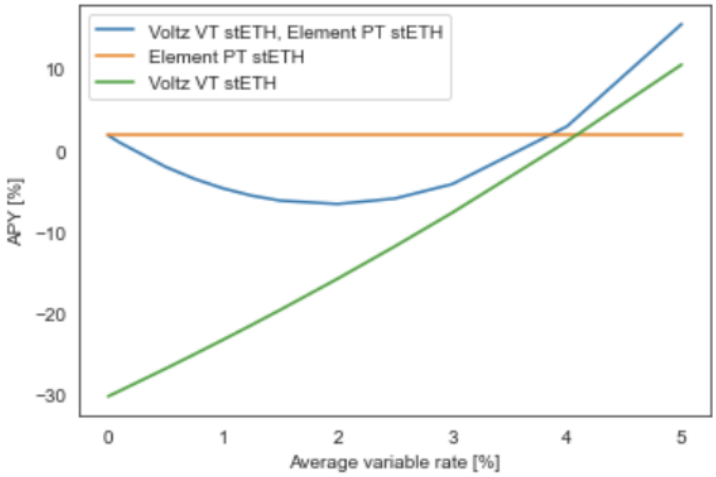


Figure 7: The projected APY for different variable rates, when hedging a Voltz VT position with stETH PTs bought from Element Finance. Both positions have similar times to maturity.

I am investigating these two signals using historical APY (variable rates) data for a given token, over some lookback window L from the current time t for entering into a position. Different lookbacks are used to infer different types of trends and information diffusion effects. If the lookback window is sufficiently small, then long-term trends are not picked up and mean-reversion effects can dominate. To capture medium-term trends in the rate, a lookback window of 30 days was considered as the trend-following window. For lower volatility assets, such as futures contracts, lookback windows of several months are not unusual. Given the higher volatility of APY in cryptocurrency markets, I selected a more short-term window of 30 days. In the 30 day window from present time t the trend is estimated as the gradient of the APY rate with respect to time over that lookback period. A variation of linear regression called RANSAC [10] was used to fit the trend, where RANSAC models have been reported to be more robust than standard linear regression against outliers in the data, which is highly relevant given the limited volume of APY lending and borrowing time series. The lookback window is rolled and updated each day, such that a new trend β_t may be calculated at each t. The "trading trend" signal, $S_{\text{trend},t}$, can be written:

$$S^{\pm}_{\text{trend},t} = \beta_t \pm \sigma_{\text{buffer,trend}}, \tag{11}$$

where we require that the learned trend be above a buffer threshold defined by a constant $\sigma_{\text{buffer,trend}}$.

The "trading rate" signal is instead based on the previous 5 days moving average, capturing shorter-term momentum in the rate. Current APY at t above these signal values is associated with high rates and high growth in rates, hence an overcorrection to the rate that will mean-revert in the short-term. In such a scenario, the strategy will short the rate for a particular token. The corresponding signal can be written iteratively:

$$S^{\pm}_{\text{rate},t} = (1-\alpha)S_{\text{rate},t-1} + \alpha \text{APY}_t \pm \sigma_{\text{buffer,rate}}. \tag{12}$$

To be more conservative, we only long or short if the rate is below the $S_{\text{rate},t}$ signal to within some buffer value, where the buffer is calculated from the standard deviation of the historical moving average in rate. Such sensible buffers prevent rapid changes in long and short positions, and therefore help to bring down gas and transaction costs. It also makes sense that we only wish to take positions in rates which we feel are confidently above the previous moving averages. Moving averages are computed using exponential weighting (by a factor α set to 0.8), such that more recent rate values are weighted more strongly in the averaging. For gas fee considerations simple moving averages might also be considered in place of the exponentially-weighted moving average. However, this is only relevant if signal calculation is performed as part of a smart contract running on-chain. The signal execution plan, as pseudocode, and performance is briefly presented in Figure 8 for trading trends and trading rates.

Infrastructure development: Voltz `Risk Engine`

In addition to strategy development at Voltz, I have been actively involved in the development and maintenance of key pieces of infrastructure covering risk management, backtesting, and live-trading/on-chain to off-chain data extraction codebases.

If $S^{+}_{\text{trend},\,t} > 0:$ LONG

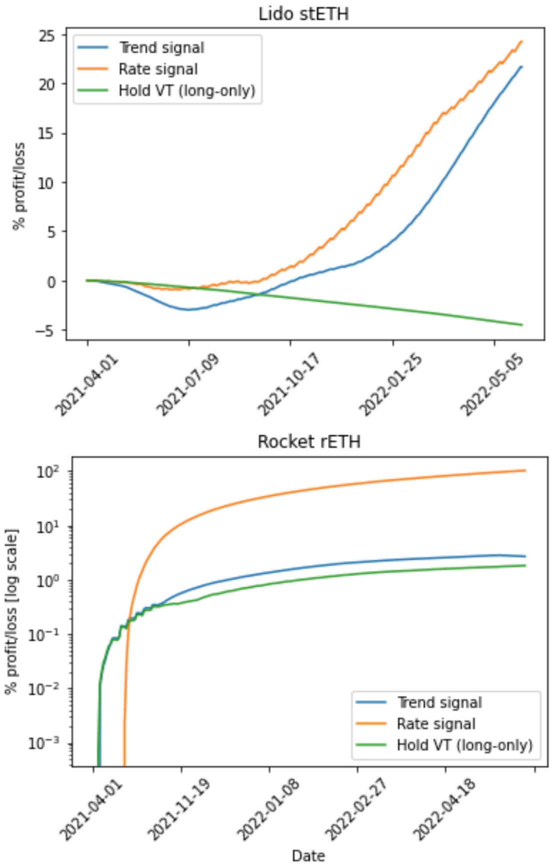
Else if $S^{-}_{\text{trend},\,t} < 0:$ SHORT

Else : EXIT position

If $\text{APY}_t > S^{+}_{\text{rate},\,t}:$ SHORT

Else if $\text{APY}_t < S^{-}_{\text{rate},\,t}:$ LONG

Else : EXIT position



A: Pseudocode for the trend and rate trading strategies

B: Equity curves for the trend (orange) and rate (blue) signals, applied to stETH and rETH variable rates

Figure 8: Performance of the trend and rate momentum strategies in the liquid staking stETH and rETH historical rates data. Trend (blue) and rate (orange) equity curves are shown, compared with the green curve which corresponds to just holding a VT position in stETH or rETH i.e just taking a long position on the variable rate.

I have developed the Voltz `Risk Engine` as the central framework for both the parameter optimisation of the Voltz IRS parameters and margin logic, and associated risk management to control the expected number of liquidatable and insolvent actors with the IS AMM. My full implemenation of the `Risk Engine` is publicly-accessable here.

The `Risk Engine` has been fully-developed and deployed to manage all positions within the AMM, and optimised for all Voltz stablecoin, stETH, and rETH IRS pools currently running on the Ethereum virtual machine. Functionality covers:

- The full simulation of an IRS pool for user-defined fixed-taker (FT), variable-taker (VT) and liquidity provider (LP) positions.
- The incorporation of different stablecoin, stETH, and rETH APY conditions, based on and bootstrapped from historical data, and calibrated with the Cox-Ingersoll-Ross (CIR) mean-reverting short-rate model [11].
- The modelling of different fixed-rate market conditions, actor leverages, and market APY volatilities, scaled and simulated under different CIR conditions.
- All protocol parameters defined for the Voltz AMM and margin calculator logic, which can be set by the user and/or optimised according to maximum leverage constrained my a minimum number of actor liquidations in the simulated IRS pool.
- The use of Optuna [12] to perform optimisation of the Voltz IRS protocol parameters and fee structure parameters, using a Bayesian Tree-structured Parzen Estimator algorithm.
- A repository of scraped historic APY data from Aave, Compound, Lido, Rocket, Euler, and other popular DeFi platforms.
- Coherent risk measure implementations for liquidation and insolvency probabilities through the Voltz `RiskMetrics` class.

Optimised simulations from the `Risk Engine` are used to ensure healthy gaps between the user initial margins deposited into the AMM, and their associated liquidation margins ("maintenance" margins). This helps to minimise liquidations for key FT, VT, and LP positions in the protocol.

Infrastructure development: Voltz `Strategy Backtesting Framework`

Another piece of infrastructure I have developed as a quant at Voltz is the `Strategy Backtesting Framework` (SBF), an event-level Python-based trading framework designed to mimic the features of a live trading system, and be adapted to work with on-chain data such as cryptocurrency prices, rates, and associated oracles. The SBF is designed to work with generic trading strategies and evaluate their historic performance. Strategies are provided as independent classes which must inherit from the `Strategy` abstract base class. The strategies are evaluated on an event-by-event basis, with the `Event` abstract base class handling the different types of trading events. The corresponding strategy portfolios are evaluated using the `Portfolio` class, which computes the relevant equity/PnL curves, Sharpe ratios, drawdowns, and drawdown durations.

I have defined four different events within this framework, each inheriting from `Event` object and mimicking the terminology of traditional event-level trading systems:

- `MarketEvent`: Handles the event of receiving new market data e.g. a fresh update from a rates database and/or rate oracle/price oracle.
- `SignalEvent`: Handles the event of sending a Signal from a Strategy object. This is received by a Portfolio object and acted upon.
- `OrderEvent`: Handles the event of sending an Order to an execution system. The order contains a token (e.g. "Aave USDC"), IRS Swap notional and a direction (long or short, corresponding to VT or FT positions respectively).
- Our in-build `ExecutionHandler` is responsible for the backtesting execution, mimicking a live trading system.
- `FillEvent`: Instantiated events of this type encapsulate the notion of a Filled IRS swap order, as returned by the Voltz protocol after executing the trade. Stores the IRS notional actually traded and at what fixed rate. In addition, stores the fees of the trade collected by liquidity providers.

The SBF is used as the framework for backtesting all Voltz strategies discussed in this document, and the public GitHub repository can be found here (note there is a separate development branch which is private). Additional features being actively incorporated into this framework include different DeFi yield curve estimates and models of slippage.

Infrastructure development: Voltz AMM and Margin Engine v2

In addition to the infrastructure work surrounding the `Risk Engine` and SBF, I am also experienced in the developments and implementation towards v2 of the Voltz protocol, including updates to the AMM and margin calculator logic. This work includes:

- investigating strategies involving perpetual swaps and hedging the associated funding rate risk.
- New margin calculation dynamics, including more efficient logic for the liquidation margin requirements.
- Cross-margining, so that actors in the protocol may work with margins calculated across several different IRS pools, pool maturities, and underlying coins and tokens.
- Margin payments across different fiat and cryptocurrencies, and models to measure the exchange rate risk and update the margins to account for this additional risk exposure.
- Generalising the margins to work with prices as well as fixed and variable rates.
- Gaussian processes as Bayesian models for calculating rate volatilities, and using this to forecast worst case APY values directly from liquidity index calculations. This is relevant for reliable liquidation margin requirements in our margin logic, and would replace the limited Cox-Ingersoll-Ross models currently used for computing the worst case deviations in APYs for different tokens.

Quantitative strategies experience at Winton Capital

Quant experience at Winton Capital focused on three areas:

- Exposure to trend-following strategies, including time-series and cross-sectional momentum, and oscillators.
- Non-linear risk measures and management in options portfolios, with futures underlyings.
- GDP and CPI announcement strategies applied to baskets of FX currency futures.

Systematic Trend-following

My experience at Winton surrounding futures included operations and exposure to systematic trend-following strategies, including momentum and oscillators. Time-series momentum signals were based on following the time series of individual commodities futures, continuously rolled as is standard with futures contracts traded at CTAs. The time series signal at time $t+1$ would be based on the sign of the returns from that particular futures contract f from the previous X months (typically 5-7, up to 12 months) to present, generating returns from t to $t+1$:

$$R^{\mathrm{TS-MOM},f}_{t,t+1} \; = \; \mathrm{sign}(R^f_{t-X,t}) \times \frac{Y\%}{\sigma^f_t} R^f_{t-X,t}, \tag{13}$$

where the future forecast volatility σ^f_t is computed from the covariance matrix over a basket of futures f, updated according to the risk system, and the position sizing is normalised to scale the volatility to a maximum annualised $Y\%$ volatility over the portfolio.

For the cross-sectional momentum signal the same form of returns equation is used, but with a weight factor where the historical return from an individual future f is compared to an equally-weighted portfolio of all futures in the universe of interest. This weighting favours futures in the basket which outperform the equally-weighted portfolio in the historical time series:

$$R^{\mathrm{XS-MOM}}_{t,t+1} = \sum_{f=1}^{N} w^f_t R^f_{t,t+1}, \tag{14}$$

where the weight is defined as

$$w_t^f = \frac{1}{N}(R^f_{t-X,t} - R^{\text{EW}}_{t-X,t}) \tag{15}$$

for some equally-weighted portfolio, EW, over the basket of N futures comprising the strategy cross-section. Here position sizing is normalised to N, rather than an annualised volatility.

Oscillator signals are formed using the crossing of different exponentially weighted moving averages on historic pricing time series. The oscillator signal is formed from the difference between moving averages over longer and shorter time-horizons, e.g. 26 and 12 days, and long and short positions are taken when this differenced average crosses above and below some average of averages for the same series i.e.

$$\text{EWMA}_f(26) - \text{EWMA}_f(12) > \text{avg}(\text{EWMA}_f) : \text{long}, \ f \text{ etc.} \tag{16}$$

Options Risk Profiles

Aside from futures I was also involved in options quant research. For my work on options, the aim was to develop a risk system that would account for expected shortfalls and maximum drawdown estimates in trading options portfolios, at that point not considered in Winton portfolios. The aim was to launch strategies trading both straddles and strangles, and also options where a position in the underlying is taken as a hedge (i.e. covered calls and protected puts). The risk system I developed was based on Monte Carlo simulations of Black-Scholes options models with a futures underlying (the Black '76 model), where the shortfall was estimated from a quadratic model of the PnL in a given time interval:

$$\text{PnL} = \tilde{\Delta}R + \frac{1}{2}R^T\tilde{\Gamma}R, \tag{17}$$

where $\tilde{\Delta}$ and $\tilde{\Gamma}$ are the simulated option Δ and Γ Greeks, in units of price. From this we developed a new non-linear estimate of the forecast volatility for the Winton portfolio, including options:

$$\sigma_P^2 = \frac{1}{2}\text{tr}[(\tilde{\Gamma}\Sigma)^2] + \tilde{\Delta}^T\Sigma\tilde{\Delta} \tag{18}$$

,
where Σ is the Winton portfolio covariance matrix, used in the original linear risk management of the futures-only universe. The second term arises from the original linear portfolio, but the first term incorporates the second-order effect of trading convexity risk with options on the futures.

This risk framework was then further extended in two ways:

- The PnL was calculated from an eigenvalue decomposition of the option Δ, Γ, and the linear covariance matrix of assets, Σ, already used in the existing Winton linear risk system:

$$\sigma_P^2 = \frac{1}{2}\sum_i \lambda^2_{\tilde{\Gamma},i}\lambda^2_{\Sigma,i} + \tilde{\Delta}^T\Sigma\tilde{\Delta} \tag{19}$$

- Gaussian mixture models were developed to model the landscape of different options strategies, at different strikes, times to maturity, and implied volatilities. Model learning and optimisation was performed using the EM algorithm [13] and the Bayesian Information Criterion. From these mixture model landscapes, different PnL profiles were projected to estimate different risk measures for use in the options trading strategy.

CPI announcements and FX Futures

The final strategy I worked on at Winton was preliminary, but the idea centred around scraping data sources for different macroeconomic variable announcements (inflation CPI, GDP, unemployment rates) in both G10 and emerging market economies. Signals were built using linear factor models based on CPI, GDP, and unemployment announcements for different countries. Long and short positions in a basket of FX currency futures were taken based on the relevant improvement in the macroeconomic announcement in month m relative to month $m-1$. For example, if the CPI announcement of a particular county (say Brazil) in February was lower than in January, then a long position would be taken in the Brazilian Real-USD FX futures over several days, where it was assumed that the information flow of the announcement was sufficiently slow for this announcement effect to not be efficiently priced into the market. This strategy was backtested with over 30 years of historical data, across many currency futures, and was found to achieve attractive Sharpe ratios (above 1).

Thoughts on future strategies

For completeness, here are a few brief words on possible future avenues of quantitative research and strategy development with applications to trading cryptocurrencies and DeFi which I feel are interesting. Detailed research, development, and backtesting is required for all of these.

- Further development of yield arbitrage and carry trading along the yield curve e.g. trading cryptocurrency zero-coupon bonds and principle tokens with the same underlying, but different maturities, as carry.
- The incorporation of perpetual swaps into the fixed-income DeFi space as a means of hedging funding rate risk. Funding rates in perps are comprised of two components: the interest rate factor calculated from the different in borrowing rates between the quoted currency and base currency in the perp underlying, and the premium due to deviations of the mark price of the perp from the bid and asking prices.
- Carry on the DeFi yield curve by calculating convexity and duration, and forming hedged positions by combining durations and convexities across different coins or protocol tokens.
- Building cross-sectional momentum strategies across baskets of cryptocurrencies.
- CNN-plus-Transformer inspired statistical arbitrage, as already discussed briefly.
- Building asset pricing models of cryptocurrencies based on estimated value, growth, market, and momentum factors, inspired by Fama-French and Carhart factor models [14, 15].
- Applications of non-linear topic models to DeFi, e.g. Latent Dirichlet Allocation, to learn hidden factors unaccounted for in traditional empirical asset pricing models.
- Convex optimisation to build an arbitrage between both rates and price trading across different CFMMs. For example, one could construct a portfolio of various token pairs trading on Uniswap v3 and Curve, together with d multi-token pools on e.g. Balancer. With this one could then perform a multi-asset constrained optimisation over these AMMs to identify mispricings between sets of tokens and trade on this accordingly.
- Building very general carry portfolios, where returns are driven by differences in borrowing/lending rates and exchange rate risk, which combine both cryptocurrencies and traditional finance FX rates.

Software strengths and weaknesses

Finally, a comprehensive summary of my current strengths and weaknesses in software skills and associated data science techniques.

Current strengths:

- Classical machine learning algorithms (Random Forest, SVM, Linear regression, robust regression, regularisation, Gaussian mixture models with scikit-learn).
- Econometrics analysis of time series, including ARMA and ARIMA approaches.
- Data science with Pandas, Numpy, Matplotlib, Statsmodels.
- Dune analytics database analysis for reading on-chain data.
- Terabyte scale data analysis with C++.
- Bash scripting.
- Python progamming.
- Deep learning ML with Tensorflow and PyTorch: CNNs, RNNs, LSTM, autoencoders.
- Ensemble learning, bagging, and boosting with XGBoost, LightGBM, and CatBoost.
- Probabilistic programming and Bayesian inference with Pyro.
- Topic modelling with Bayesian techniques, including Latent Dirichlet Allocation.
- Hierarchical Bayesian models with Markov Chain Monte Carlo and Variational inference.

- Developing skills in smart contract programming and cryptography (e.g. Merkle tree construction and hashing) with Solidity and TypeScript.
- Cloud computing with AWS, which I want to develop further.
- Grid computing at the Large Hadron Collider, relevant for terabyte scale data analysis.
- GPUs and Cuda for machine learning.
- Package and environment management with Pip and Anaconda.
- Version control with GitHub, GitLab, and BitBucket.

Current weaknesses:

- Limited experience with SQL and MongoDB; would like to develop skills here.
- No experience in Rust and Go, very keen like to develop skills there as appropriate.
- Limited experience with latest techniques in NLP, such as transformers, but I intend to acquire this expertise as it becomes increasingly relevant in quant research.

References

[1] Adams, H. *et al* (2021). Uniswap v3 Core. https://uniswap.org/whitepaper-v3.pdf

[2] Dickey, D. A., & Fuller, W. A. (1979). Distribution of the Estimators for Autoregressive Time Series With a Unit Root. *Journal of the American Statistical Association*, 74(366), 427–431. https://doi.org/10.2307/2286348

[3] Pei, Y. *et al* (2017). An Elementary Introduction to Kalman Filtering. https://doi.org/10.48550/arXiv.1710.04055

[4] Johansen, S. (1991). Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models. *Econometrica*, 59(6), 1551-1580. https://doi.org/10.2307/2938278

[5] Barker, A. L. *et al* (1995). Bayesian Estimation and the Kalman Filter. *Computers Math. Applic.* Vol. 30, No. 10, pp. 55-77. https://doi.org/10.1016/0898-1221(95)00156-S

[6] Johansen, S. (1995). Likelihood-Based Inference in Cointegrated Vector Autoregressive Models. *Oxford Academic*. https://doi.org/10.1093/0198774508.001.0001

[7] Guijarro-Ordonez J., Pelger, M., Zanotti, G. (2022). Deep Learning Statistical Arbitrage. https://doi.org/10.48550/arXiv.2106.04028

[8] Nelson, C., and Siegel, A. F. (1987). Parsimonious Modeling of Yield Curves. *Journal of Business*, Vol. 60, 473-489.

[9] Pedersen, L. H. (2015). Efficiently Inefficient: How Smart Money Invests and Market Prices Are Determined. *Princeton University Press.* https://doi.org/10.2307/j.ctt1287knh

[10] Fischler, M. A. & Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. ACM.* 24 (6): 381–395. doi:10.1145/358669.358692

[11] Cox, J.C., Ingersoll, J. E., & Ross, S. A. (1985). A Theory of the Term Structure of Interest Rates. *Econometrica.* 53 (2): 385–407. doi:10.2307/1911242. JSTOR 1911242.

[12] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019) Optuna: A Next-generation Hyperparameter Optimization Framework. *KDD.* https://arxiv.org/abs/1907.10902

[13] Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B.* 39 (1): 1–38. JSTOR 2984875. MR 0501537.

[14] Fama, E. F. French, K. R. (2014). A Five-Factor Asset Pricing Model. *Fama-Miller Working Paper*, SSRN: http://dx.doi.org/10.2139/ssrn.2287202

[15] Carhart, M. M. (1997) On Persistence in Mutual Fund Performance. *The Journal of Finance*, 52(1), 57-82. https://doi.org/10.1111/j.1540-6261.1997.tb03808.x