# Multivariate Volatility Forecast and Multi-asset Indifference Pricing

team@vest.xyz

March 2023

**Introductory Notes**

What follows below is research carried out by Vest Labs with the primary intent of it being discussed on the labs' forum. This particular session can be found here: https://research.vest.xyz/t/multivariate-volatility-forecasting-for-risk-indifference-pricing/195

## Motivation

In our previous write-up "Bandit Funding Rates", we explored a new mechanism design for setting funding rates on perpetual futures exchanges. However, we observed that funding rates would sometimes spike up unexpectedly during simulation runs, which would not be ideal for traders. This in part can be due to the fact that the bandit framework tried to manage the default risk for the exchange solely by switching funding rates and did not consider charging spread (or equivalently, charging mark price different from the index).

In order to determine a fair price for each contract, we explore risk indifference pricing where the risk is defined as the possible loss the exchange is exposed to when traders are net profitable. We use entropic value-at-risk (EVaR) to quantify such risk given the state of the exchange (long and short open interests). Since net trader PnL is a function of the prices of the underlying assets, calculating entropic value-at-risk requires some estimate of the distribution of future returns on these assets. Below we explore multiple models for forecasting multivariate volatility and discuss how the forecasted covariance matrices can be used for indifference pricing.

## Multivariate Volatility Forecasting Models

We aim to forecast the volatility of $n \geq 50$ crypto assets. Approaches such as stochastic volatility models or copulas seem too complex for such high dimensions. Alternatively, neural network based methods could be too "black box" for a decentralized protocol to use. Instead, we choose to focus on multivariate GARCH (mGARCH) models and make the following simplifying assumptions: the log returns of all price processes follow a multivariate normal distribution and have zero mean.

More precisely, let $\{\mathbf{x_t}\} \in \mathbb{R}^n$ be a vector stochastic process of standardized log returns. Then, our model can be represented as

$$\mathbf{x_t} \mid \mathcal{F}_{t-1} \sim \mathcal{N}(0, \mathbf{V_t})$$

where $\mathcal{F}_{t-1}$ is the information up to time $t-1$ and $\mathbf{V_t}$ is a (positive semi-definite) covariance matrix.

We compare two popular classes of mGARCH models as well as another class based on hierarchical clustering that we conjecture is a reasonable parametric design for modelling a large number of diverse crypto assets. In this paper, we focus on GARCH models of order $(1,1)$ instead of generalized orders $(p,q)$.

Our benchmark model will be the univariate GARCH$(1,1)$ model fit independently to each asset, hence assuming zero correlation between assets. That is, $\mathbf{V_t} = diag(\sigma^2_{x_{t,1}}, \ldots, \sigma^2_{x_{t,n}})$ where $\sigma^2_{x_{t,i}} = \omega_i + \alpha_i x^2_{t-1,i} + \beta_i \sigma^2_{x_{t-1,i}} \forall 1 \leq i \leq n$ fit to historical standardized log returns $\mathbf{X}$, a matrix of dimension $T \times n$. This requires an estimation of $3n$ parameters.

## Factor-ARCH

Factor-ARCH models assume that there exist unobservable factors $\mathbf{f}_t$ that are conditionally heteroscedastic and generate the log returns $\mathbf{x}_t$ for a set of assets. That is,

$$\mathbf{x}_t = \mathbf{Z}\mathbf{f}_t$$

where $\mathbf{Z}$ is a time-invariant linear map.

Orthogonal-GARCH (O-GARCH) [Ale02] considers the eigendecomposition of the covariance matrix

$$\mathbf{V}_t = \mathbf{Q}\mathbf{\Lambda}^2\mathbf{Q}^T \approx \tilde{\mathbf{Q}}\tilde{\mathbf{\Lambda}}^2\tilde{\mathbf{Q}}^T$$

where the diagonal of $m$ largest eigenvalues $\tilde{\mathbf{\Lambda}}^2 = diag(\sigma^2_{f_{t,1}}, \dots, \sigma^2_{f_{t,m}})$ represent the conditional variances of each factor that can be modeled by univariate GARCH. Here, $\mathbf{Z} = \tilde{\mathbf{Q}}$ is an $n \times m$ (semi-)orthogonal matrix. We choose $m$ based on the Marchenko-Pastur law from Random Matrix Theory, where any eigenvalue $\lambda < \lambda_+ = (1 + \sqrt{n/T})^2$ can be assumed to be associated with noise [LCBP99].

Generalized Orthogonal-GARCH (GO-GARCH) [VdW02] relaxes the assumption of (semi-)orthogonality by allowing $\mathbf{Z}$ to be (left) invertible but not necessarily orthogonal. GO-GARCH parameterizes $\mathbf{Z}$ with singular value decomposition:

$$\mathbf{V}_t = \mathbf{Z}\mathbf{D}_t\mathbf{Z}^T \approx \tilde{\mathbf{Z}}\tilde{\mathbf{D}}_t\tilde{\mathbf{Z}}^T,\ \tilde{\mathbf{Z}} = \tilde{\mathbf{Q}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^T$$

where $\mathbf{Q}$ and $\mathbf{\Lambda}$ are defined as above and $\tilde{\mathbf{D}}_t = diag(\sigma^2_{f_{t,1}}, \dots, \sigma^2_{f_{t,m}})$ is a diagonal matrix of the conditional variances of each factor modeled by univariate GARCH. This requires an estimation of $\frac{(2n-m+5)m}{2}$ parameters: $3m$ parameters for each univariate GARCH and $\frac{(2n-m+1)m}{2}$ parameters for $\tilde{\mathbf{U}}$. In practice, $\tilde{\mathbf{U}}$ is estimated with Independent Component Analysis [Gha19]. Note that O-GARCH with $m = n$ corresponds to $\mathbf{U} = \mathbf{I}_n$.

## Conditional Correlation

Conditional correlation models estimate conditional variance and correlation separately. Constant Conditional Correlation-GARCH (CCC-GARCH) [Bol90] is defined as:

$$\mathbf{V}_t = \mathbf{D}_t\mathbf{R}\mathbf{D}_t$$

where $\mathbf{D}_t = diag(\sigma_{x_t,1}, \dots, \sigma_{x_t,n})$ is a diagonal matrix of the conditional standard deviations of each asset, which can be modeled by univariate GARCH, and $\mathbf{R}$ is the constant conditional correlation matrix.

Dynamic Conditional Correlation-GARCH (DCC-GARCH) [EIS01] relaxes the constant conditional correlation assumption by modelling the time-varying conditional correlation as:

$$\mathbf{V}_t = \mathbf{D}_t\mathbf{R}_t\mathbf{D}_t$$

where $\mathbf{D}_t$ is defined as above, and $\mathbf{R}_t = (1 - \alpha - \beta)\mathbf{R} + \alpha(\mathbf{x}_{t-1}\mathbf{x}^T_{t-1}) + \beta\mathbf{R}_{t-1}$, where $\mathbf{R}$ is the unconditional correlation matrix and $\alpha$ and $\beta$ are the parameters to be estimated. This method requires an estimation of $\frac{(n+1)(n+4)}{2}$ parameters: $3n$ parameters for each univariate GARCH, $\frac{n(n-1)}{2}$ parameters for unconditional correlation matrix $\mathbf{R}$, and 2 additional parameters for dynamic model $\alpha$ and $\beta$.

## H-GARCH

Hierarchical-GARCH (H-GARCH) [Ave20] is a class of mGARCH models that begins by forming hierarchical clusters of assets and reconstructs a covariance matrix by estimating separately the covariance within each cluster and between clusters. H-GARCH provides a good parametric method for modeling crypto classes that may differ in liquidity and observation frequency.

We perform hierarchical clustering using Ward's minimum variance method with a proper distance metric $d_{ij} = \sqrt{\frac{1}{2}(1 - \rho_{ij})}$, where $\rho_{ij}$ is the correlation between assets $i$ and $j$.

Intuitively, we want to apply GO-GARCH to recover within-cluster covariance since clusters should contain relatively correlated assets. DCC-GARCH can then recover the between-cluster covariance.

For example, for two clusters, let $\mathbf{AD}_{t,1}\mathbf{A}^T$ and $\mathbf{BD}_{t,2}\mathbf{B}^T$ be the within-cluster covariance matrices recovered from GO-GARCH for clusters 1 and 2 respectively and $\mathbf{C}_{1,2} = cov(\mathbf{f}_{t,1}, \mathbf{f}_{t,2})$ be the covariance between the factors of cluster 1 and 2 recovered from DCC-GARCH. Then, the full covariance matrix is:

$$\begin{bmatrix} \mathbf{AD}_{t,1}\mathbf{A}^T & \mathbf{AC}_{1,2}\mathbf{B}^T \\ (\mathbf{AC}_{1,2}\mathbf{B}^T)^T & \mathbf{BD}_{t,2}\mathbf{B}^T \end{bmatrix}$$

Extension to more than two clusters is straight forward. Note, however, that this method does not guarantee the output to be positive semi-definite (which means that the output may not be a valid covariance matrix). Applying the proof in [Ale02], we can guarantee positive semi-definiteness by rounding correlation between each pair of factors to

$$corr(f_{t,i}^{(k)}, f_{t,j}^{(l)}) \leftarrow \begin{cases} -c & \text{if } corr(f_{t,i}^{(k)}, f_{t,j}^{(l)}) < -c \\ corr(f_{t,i}^{(k)}, f_{t,j}^{(l)}) & \text{if } -c \leq corr(f_{t,i}^{(k)}, f_{t,j}^{(l)}) < c \\ c & \text{otherwise} \end{cases}$$

where $c = \sqrt{\frac{1}{(m-m_i)(m-m_j)}}$, $m_i$ and $m_j$ are number of factors for cluster $i$ and $j$ respectively, and $m = \sum_{i \in clusters} m_i$. Another approach is to modify the number of factors for each cluster until the output is positive semi-definite, but this is not explored in this work.

# Method

## Experiment Data

We fitted all models using Binance US 5-minute frequency times series data over 3 days (864 observations) of the top 50 crypto assets in market capitalization. To robustly test for forecast accuracy, we chose 3 periods of price paths that correspond to prior, during, and post USDC depeg situation in early March 2023 (see Appendix A).

## Performance Measure

We compare the performance of each model using Model Confidence Set [HLN03] (MCS) method. The MCS sequence of tests delivers the Superior Set of Models (SSM) that has Equal Predictive Ability (EPA) in terms of a specified loss function constructed with respect to desired model characteristics. The MCS procedure achieves SSM through sequential testing procedure which eliminates at each step the worst model, until the hypothesis of EPA is accepted for all the models belonging to the SSM. We use the $T_{max}$ null hypothesis for EPA. For details of the implementation, refer to [BC14].

Instead of using loss functions MAE, MSE, or the Frobenius norm to only compare the forecasted covariance against estimated covariance, we use a continuous ranked probability score called the sample energy score (ES) [JKL18], where given a forecasted distribution $F$ on $\mathbb{R}^n$, i.i.d samples $\mathbf{X}_1, \ldots, \mathbf{X}_k \sim F$, and realized observation $\mathbf{y}$,

$$ES(F, \mathbf{y}) = \frac{1}{k} \sum_{i=1}^{k} ||\mathbf{X}_i - \mathbf{y}||_2 + \frac{1}{2k^2} \sum_{i=1}^{k} \sum_{j=1}^{k} ||\mathbf{X}_i - \mathbf{X}_j||_2$$

We perform the MCS procedure on ES loss for each model with 100 rolling outputs of 5-minute and 1-hour ahead covariance forecasts. Since all models are fitted to $\delta t$ = 5-minute frequency data and have a 1-step ahead analytical forecast $\widehat{\mathbf{V}}_{t+\delta t}$, the $\tau$-ahead covariance is approximated by $\widehat{\mathbf{V}}_{t+\tau} \approx \frac{\tau}{\delta t} \widehat{\mathbf{V}}_{t+\delta t}$. We estimate realized covariance by the sum of squared returns $\mathbf{V}_{t+\tau} \approx \sum_{s=1}^{\tau/\delta t} \mathbf{x}_{t+s\delta t} \mathbf{x}_{t+s\delta t}^T$.

### Model Specifications

All GARCH models (and DCC) are of order $(1,1)$ and we denote by GO-GARCH-MP as GO-GARCH performed while keeping $m < n$ factors using the result from Marchenko-Pastur (MP) law. H-GARCH will use MP as default, while GO-GARCH without MP will keep all $n$ factors.

Table 1: Set of parameters used for the simulation

| Parameter | Symbol | Value |
|---|---|---|
| Sample size | $T$ | 864 |
| Sample frequency | $\delta t$ | 5 min |
| Number of clusters | $n_c$ | 2 |
| ES sample size | $k$ | 500 |
| MCS confidence level | $1 - \alpha$ | 0.9 |

## Results

For the 5-minute ahead forecast, Univariate GARCH and DCC-GARCH performed the best while GO-GARCH-MP performed the worst. However, for the 1-hour ahead forecast, both GO-GARCH-MP and H-GARCH consistently outperformed all other models, suggesting that using the Marchenko-Pastur law to reduce the number of factors could effectively denoise the input data and increase forecast accuracy for longer-term covariance. This also indicates that both Univariate GARCH and DCC-GARCH are overfitting.

We also observed that DCC-GARCH takes the longest time to run, followed by H-GARCH, GO-GARCH, and GO-GARCH-MP, due to the number of parameters required for each model. In practice, since we expect to use forecasts on the order of hours-ahead and we want to optimize for speed, GO-GARCH-MP seems to be the best model that fits our needs as of now. It remains to be shown if H-GARCH could result in similar performance if it alternatively used GO-GARCH-MP instead of DCC-GARCH as the subroutine for estimating between-cluster covariances.

Table 2: Mean ES loss for 5-minute ahead covariance forecast

| Model | Mean ES (Period 1) | Mean ES (Period 2) | Mean ES (Period 3) |
|---|---|---|---|
| Univariate GARCH | **0.008475** | 0.02025 | **0.014898** |
| DCC-GARCH | **0.008476** | **0.020077** | **0.01476** |
| GO-GARCH | **0.008609** | **0.020209** | 0.015005 |
| GO-GARCH-MP | 0.008899 | 0.021385 | 0.015305 |
| H-GARCH | **0.008654** | 0.020821 | **0.014975** |

* Bold indicates inclusion in the Superior Set of Models.

Table 3: Mean ES loss for 1-hour ahead covariance forecast

| Model | Mean ES (Period 1) | Mean ES (Period 2) | Mean ES (Period 3) |
|---|---|---|---|
| Univariate GARCH | 0.016005 | 0.03277 | 0.028248 |
| DCC-GARCH | 0.015811 | 0.032043 | 0.027385 |
| GO-GARCH | 0.016873 | 0.034644 | 0.029975 |
| GO-GARCH-MP | **0.00987** | **0.025829** | **0.019445** |
| H-GARCH | 0.010548 | 0.027316 | 0.021135 |

* Bold indicates inclusion in the Superior Set of Models.

# Extension to Risk Indifference Pricing

As mentioned in Motivation, our goal is to derive a risk indifferent price under multi-asset trading. With a few additional assumptions on the underlying process, we obtain a closed-form solution for pricing any combination of assets.

Given a forecast of $n \times n$ covariance matrix $\mathbf{V}_{t+1}$ of conditional log-returns, under the zero mean/drift assumption, we can recover the mean of conditional returns as $\mathbf{1}$ and the covariance matrix as $\mathbf{V}^r_{t+1} = \exp(\mathbf{V}_{t+1}) - I$. As returns are calculated with respect to the current prices $\mathbf{p}_t = (p_{1,t}, \cdots, p_{n,t})^T$, the mean of conditional prices is $\mathbf{p_t}$ and the covariance matrix is $\mathbf{V}^p_{t+1} = \text{diag}(\mathbf{p}_t)\mathbf{V}^r_{t+1}\text{diag}(\mathbf{p}_t)^T$.

We now approximate the conditional price $\mathbf{p}_{t+1}$ with multivariate normal distribution $\mathcal{N}(\mathbf{p}_t, \mathbf{V}^p_{t+1})$, matching the first two moments. As any portfolio value $X_{t+1}(\mathbf{q})$ is a linear transformation on the price vector $\mathbf{q}\mathbf{p}_{t+1}$ with a vector of signed position sizes $\mathbf{q} = (q_1, \cdots, q_n)$, the above approximation allows us to model $X_{t+1}(\mathbf{q})$ as a random variable following univariate normal distribution $N(\mathbf{q}\mathbf{p}_t, \mathbf{q}\mathbf{V}^p_{t+1}\mathbf{q}^T)$. If we consider $\mathbf{q}$ as the net trader positions for $n$ markets, the liability of the exchange can then be considered as $X_{t+1}(\mathbf{q} - C$, where $C$ consists of the trader collateral and any additional liquidity. This allows us to use a nice formula for entropic value-at-risk: $EVaR_{1-\alpha}(X_{t+1}(\mathbf{q}) - C) = \mathbf{q}\mathbf{p}_t + \sqrt{-2(\mathbf{q}\mathbf{V}^p_{t+1}\mathbf{q}^T)\ln\alpha} - C$.

Using the approximated $EVaR_{1-\alpha}(X_{t+1}(\mathbf{q}) - C)$, we can define the premium for any new trade that changes net open interest from $\mathbf{q}$ to $\mathbf{q'}$ as

$$\pi(\mathbf{q}, \mathbf{q'}) = EVaR_{1-\alpha}(X_{t+1}(\mathbf{q'}) - C') - EVaR_{1-\alpha}(X_{t+1}(\mathbf{q}) - C).$$

Given the cash-invariance property of entropic value-at-risk, this is equivalent to

$$EVaR_{1-\alpha}(X_{t+1}(\mathbf{q}) - C + \pi(\mathbf{q}, \mathbf{q'})) = EVaR_{1-\alpha}(X_{t+1}(\mathbf{q'}) - C'),$$

which is the invariant we want to maintain.

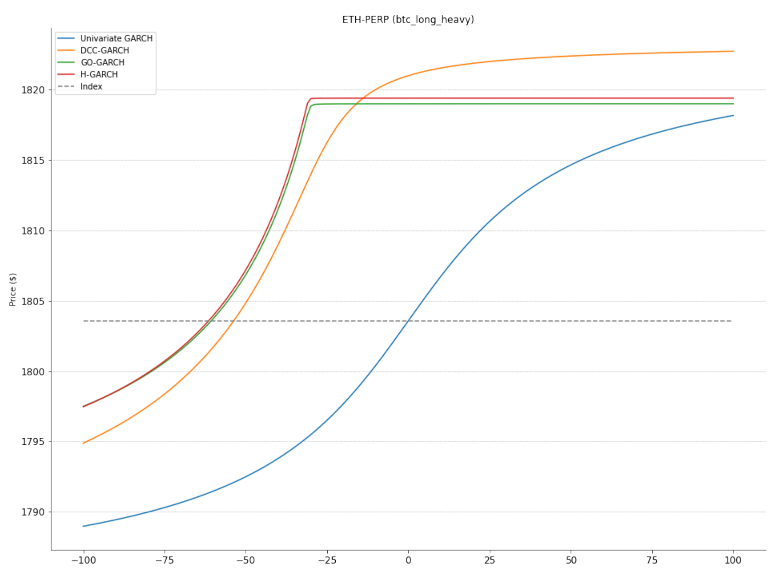


Figure 1: ETH-PERP price when BTC-PERP market is long heavy

Figure 1 is an example of mark price for ETH-PERP (calculated as index + (premium / size)) when BTC-PERP market has net 2 long OI. We can see that under Univariate GARCH, the correlation between ETH

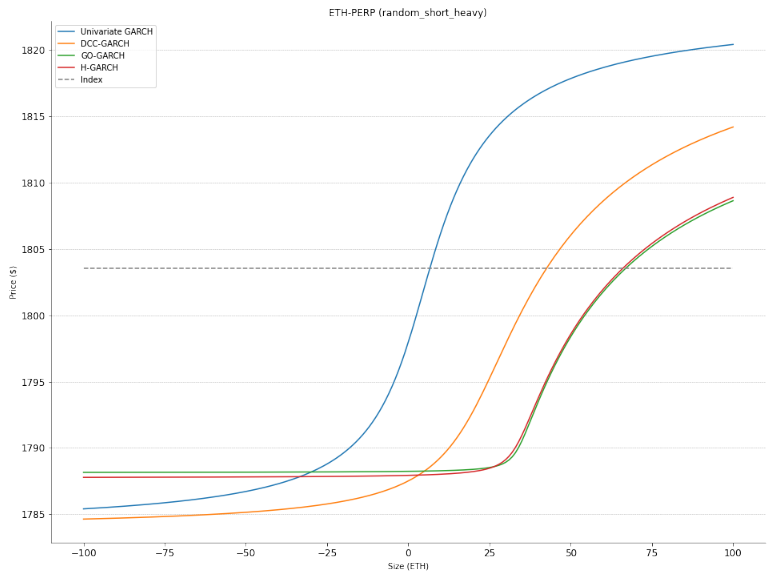


Figure 2: ETH-PERP price when most markets are short heavy

price and BTC price are completely ignored (long and short premium are priced symmetrically), where as the other models consider the positive correlation. This is reflected in higher premia for long positions and rebate for short positions up to 50 - 75 ETH. Interestingly, pricing under GO-GARCH and H-GARCH seem to converge faster to lower price for long positions, whereas pricing under DCC-GARCH seem to converge slower to higher price. Figure 2 was produced by randomly generating OI for each of the 50 markets with higher probability of being short-heavy. We find that diversifying the market portfolio still preserves the skewness of the price curve.

# Future Work

In this section, we propose several directions for future research. Firstly, other methods of hierarchical clustering as well as choosing the number of clusters should be explored. One method could be to incorporate economic theory into the cluster structure [LdP19]. This hybrid method may be useful when dealing with larger numbers of assets from different asset classes.

The choice of factors to keep for both the GO-GARCH-MP and H-GARCH models could also be improved. While the current use of the Marchenko-Pastur law has demonstrated good performance, we should consider artefacts in the data like auto-correlation or cross-sectional correlations and adjust our eigenvalue clipping thresholds accordingly [LdP16].

Finally, Kalman Filters could serve as either a method to denoise the sample covariance matrix or as an alternative to multivariate GARCH models.

# Appendix A Price data

## A.1 Crypto assets

The following Binance US markets were used as price data (sorted descending in market capitalization). Stablecoin markets were not used.

BTC/USD, ETH/USD, BNB/USD, ADA/USD, DOGE/USD, MATIC/USD, SOL/USD, DOT/USD, LTC/USD, TRX/USD, AVAX/USD, UNI/USD, LINK/USD, ATOM/USD, ETC/USD, BCH/USD, XLM/USD, FIL/USD, APT/USD, LDO/USD, HBAR/USD, QNT/USD, NEAR/USD, VET/USD, ALGO/USD, ICP/USD, APE/USD, FTM/USD, GRT/USD, EOS/USD, SAND/USD, IMX/USD, AAVE/USD, MANA/USD, EGLD/USD, XTZ/USD, FLOW/USD, THETA/USD, AXS/USD, SNX/USD, NEO/USD, OP/USD, CRV/USD, DASH/USD, CHZ/USD, MKR/USD, RNDR/USD, PAXG/USD, ZIL/USD, 1INCH/USD

## A.2 Periods

Figure 3 shows the three periods of price data used in the experiment.

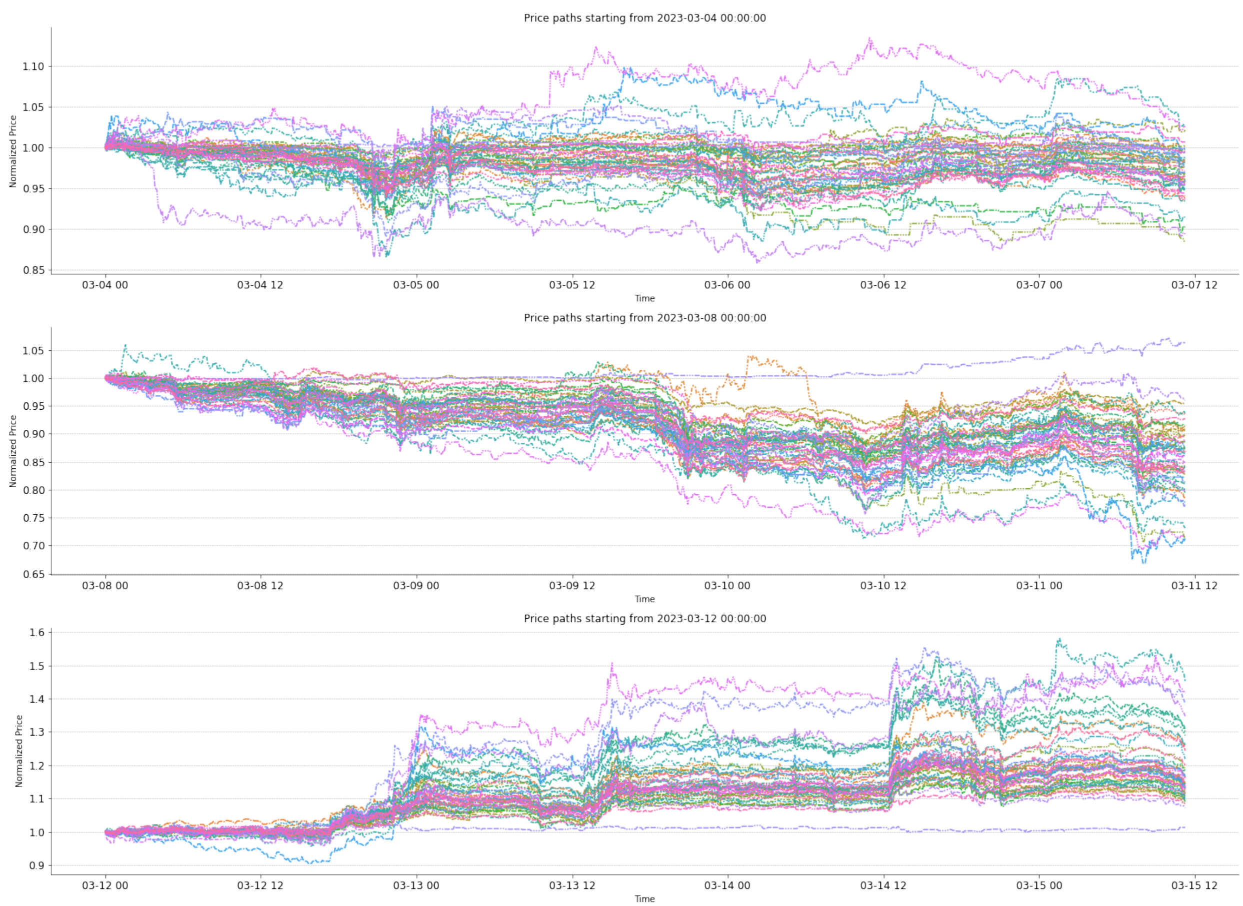


Figure 3: Normalized price data of Period 1 (🦀), 2 (🐻), 3 (🐂)

# References


[Ale02] Carol Alexander. Principal component models for generating large garch covariance matrices. *Economic Notes*, 31(2):337–359, 2002.

[Ave20] Marco Avellaneda. Hierarchical pca and applications to portfolio management. *Revista mexicana de economía y finanzas*, 15(1):1–16, 2020.

[BC14] Mauro Bernardi and Leopoldo Catania. The model confidence set package for r, 2014.

[Bol90] Tim Bollerslev. Modelling the coherence in short-run nominal exchange rates: a multivariate generalized arch model. *The review of economics and statistics*, pages 498–505, 1990.

[EIS01] Robert F Engle III and Kevin Sheppard. Theoretical and empirical properties of dynamic conditional correlation multivariate garch, 2001.

[Gha19] Alexios Ghalanos. The rmgarch models: Background and properties.(version 1.3-0). *Technical Report*, 2019.

[HLN03] Peter Reinhard Hansen, Asger Lunde, and James M Nason. Choosing the best volatility models: the model confidence set approach. *Oxford Bulletin of Economics and Statistics*, 65:839–861, 2003.

[JKL18] Alexander Jordan, Fabian Krüger, and Sebastian Lerch. Evaluating probabilistic forecasts with scoringrules, 2018.

[LCBP99] Laurent Laloux, Pierre Cizeau, Jean-Philippe Bouchaud, and Marc Potters. Noise dressing of financial correlation matrices. *Physical review letters*, 83(7):1467, 1999.

[LdP16] Marcos Lopez de Prado. A robust estimator of the efficient frontier. *Available at SSRN 3469961*, 2016.

[LdP19] Marcos Lopez de Prado. Estimation of theory-implied correlation matrices. *Available at SSRN 3484152*, 2019.

[VdW02] Roy Van der Weide. Go-garch: a multivariate generalized orthogonal garch model. *Journal of Applied Econometrics*, 17(5):549–564, 2002.