Analysis of Cryptocurrency Returns Using Quantile & Expectile Based Copula HMM

Adrija Saha(MD2203) & Shrayan Roy(MD2220), M.Stat 2nd Year

Contents

Introduction **2**

Background **2**

Objective **2**

Data Description **2**

Exploratory Data Analysis **4**

Fiting The Proposed Model **6**

Results **7**

$\tau_i = 0.05$, $\nu_k = 5$ and $K = 2$ 7

$\tau_i = 0.05$, $\nu_k = 5$ and $K = 3$ 10

$\tau_i = 0.50$, $\nu_k = 5$ and $K = 3$ 11

$\tau_i = 0.95$, $\nu_k = 5$ and $K = 3$ 12

Observations 12

Quantile and Expectile Copula Based Hidden Semi-Markov Model **13**

Appendix - Codes **15**

CQHMM Model 15

CEHMM Model 19

References **21**

Introduction

The cryptocurrency market, characterized by its dynamic nature and inherent volatility, has become a focal point for researchers seeking to understand its complex dynamics and interactions with traditional financial assets. Building upon the theoretical framework introduced in the paper titled *Quantile and expectile copula-based hidden Markov regression models for the analysis of the cryptocurrency market* (https://arxiv.org/abs/2307.06400), this extended assignment aims to implement and apply the proposed models to a real-world dataset.

Background

The cryptocurrency market, with the emergence of various crypto assets following Bitcoin's trajectory, has experienced significant fluctuations, notably during the 2017 boom and subsequent 2018 crash. The COVID-19 pandemic in 2020 further heightened market volatility, raising questions about the behavior of cryptocurrencies during crises and their impact on global financial markets.

While existing literature has explored aspects such as long memory, efficiency, and hedging properties of cryptocurrencies, the paper under consideration introduces a novel approach. It employs hidden Markov regression models and state-dependent elliptical copulas to capture unobserved heterogeneity and evolving dependency structures in cryptocurrency returns.

Objective

Here we aim to implement the *Copula Quantile Hidden Markov Model* (CQHMM) and *Copula Expectile Hidden Markov Model* (CEHMM) on a real dataset. By doing so, we seek to analyze the dynamic relationships among multiple cryptocurrency returns, considering the evolving market conditions and dependencies introduced by the state-dependent copulas.

Data Description

Following the setup of model described in the original paper. We will need data on return of cryptocurrencies and also data on some fixed covariates. Here we want to see the interaction of cryptocurrencies market and traditional assests, like - Stocks,Bonds and Indices. So we will consider some representatives of the traditional assests. Fitting a copula based regression model will help us to understand the significance of traditional assets on cryptocurrencies. Let us see the different cryptocurrencies available in this date. We will use `crypto2` package in R for extracting cryptocurrency data. The below code extracts all currently active cryptocurrencies.

```
active_list <- crypto_list(only_active = TRUE)
coin_list_2022 <- active_list %>% dplyr::filter(first_historical_data<="2022-12-31",
                                                 last_historical_data>="2022-01-01")
head(coin_list_2022,7)
```

```
## # A tibble: 7 x 8
##      id  rank name        symbol slug        is_active first_historical_data
##   <int> <int> <chr>       <chr>  <chr>           <int> <date>
## 1     1     1 Bitcoin     BTC    bitcoin             1 2010-07-13
```

```
## 2     2    18 Litecoin    LTC    litecoin            1 2013-04-28
## 3     3   715 Namecoin    NMC    namecoin            1 2013-04-28
## 4     4  6059 Terracoin   TRC    terracoin           1 2013-04-28
## 5     5   866 Peercoin    PPC    peercoin            1 2013-04-28
## 6     6  5886 Novacoin    NVC    novacoin            1 2013-04-28
## 7     8  1784 Feathercoin FTC    feathercoin         1 2013-05-03
## # i 1 more variable: last_historical_data <date>
```

We will work with top five cryptocurrencies(top w.r.t ranking) i.e. Bitcoin, XRP, Tether USDt, Ethereum and BNB, as we can see below.

```
#First 5 ranking bitocins
ranker_list <- coin_list_2022[coin_list_2022$rank%in%c(1,2,3,4,5),]
ranker_list
```

```
## # A tibble: 5 x 8
##      id  rank name        symbol slug     is_active first_historical_data
##   <int> <int> <chr>       <chr>  <chr>        <int> <date>
## 1     1     1 Bitcoin     BTC    bitcoin          1 2010-07-13
## 2    52     5 XRP         XRP    xrp              1 2013-08-04
## 3   825     3 Tether USDt USDT   tether           1 2015-02-25
## 4  1027     2 Ethereum    ETH    ethereum         1 2015-08-07
## 5  1839     4 BNB         BNB    bnb              1 2017-07-25
## # i 1 more variable: last_historical_data <date>
```

```
## > Scraping historical crypto data
## > Processing historical crypto data
```

For these cryptocurrencies, we will consider daily data from 25th July,2017 to 31st October,2023. Since, Crypto market is open 24×7. So, we have data for all these dates. We will consider the data on S&P 500, USDX, WTI Crude Oil and Gold. For extracting data on these traditional assets, we will use `tidyquant` and `quantmod` packages of R. The below code extracts data on these assets.

```
symbols <- c("^GSPC", "DX-Y.NYB", "CL=F", "GC=F")
start_date <- "2017-07-25";end_date <- "2023-10-31"
getSymbols(symbols, src = "yahoo", from = start_date, to = end_date,
           auto.assign = TRUE)
```

```
## [1] "GSPC"     "DX-Y.NYB" "CL=F"     "GC=F"
```

We note that like crypto market, stock market is not always open. So we will consider the common times points only. For a fixed time point t, we denote the data on cryptocurrencies by the following vector $\boldsymbol{Y_t} = (Y_{t1}, Y_{t2}, Y_{t3}, Y_{t4}, Y_{t5})^T$. Where, $Y_{t1}, Y_{t2}, Y_{t3}, Y_{t4}$ and Y_{t5} respectively denote the return of Bitcoin,Ethereum,USDT,BNB and XRP at time t and data on fixed covariate (fixed because we know it at time t before observing $\boldsymbol{Y_t}$) by $\boldsymbol{x_t} = (1, x_{t1}, x_{t2}, x_{t3}, x_{t4})$. Where, x_{t1}, x_{t2}, x_{t3} and x_{t4} respectively denote the return of S&P 500, USDX, WTI Crude Oil and Gold at time point t-1 i.e. previous time point. We will model the joint distribution of $\boldsymbol{Y_t}$ by eliptical copulas, Like - Normal, t Copula and will model the quantile or expectile by the $\boldsymbol{x_t}$.

After doing necessary data cleaning and adjustments we get the following data frame.

```
## 'data.frame':    1576 obs. of  10 variables:
##  $ t       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ BTC     : num  -0.0183 0.0563 0.0514 0.0236 -0.0546 ...
##  $ ETH     : num  -0.01334 0.00179 -0.05479 0.05565 0.11232 ...
##  $ USDT    : num  -0.002848 0.000321 0.004678 -0.001018 -0.004097 ...
##  $ BNB     : num  -0.00691 0.02472 -0.03406 0.00176 -0.04205 ...
##  $ XRP     : num  -0.02404 -0.0021 -0.04335 0.00873 0.05775 ...
##  $ X.GSPC  : num  0.000283 -0.000973 -0.001341 -0.000728 0.002449 ...
##  $ DX.Y.NYB: num  -0.00404 0.00203 -0.00639 -0.00429 0.00183 ...
##  $ CL.F    : num  0.01796 0.00595 0.01366 0.00925 -0.02013 ...
##  $ GC.F    : num  -0.00216 0.00849 0.00699 -0.00142 0.00474 ...
```

second to sixth columns give return data on cryptocurrencies and the last four columns give return data on traditional assets.

Exploratory Data Analysis

In the context of implementing Copula Quantile Hidden Markov Models (CQHMM) and Copula Expectile Hidden Markov Models (CEHMM) on cryptocurrency returns, Exploratory Data Analysis (EDA) plays a pivotal role. EDA is essential for understanding the inherent characteristics of the dataset, identifying patterns, and recognizing potential challenges. Through graphical and statistical techniques, EDA facilitates the identification of outliers, trends, and distributions within cryptocurrency returns. Moreover, EDA provides insights into the temporal dynamics and relationships among different cryptocurrencies, aiding in the formulation of informed hypotheses for the subsequent model implementation. It serves as a crucial preliminary step, guiding researchers in tailoring the models to capture the nuanced behaviors of cryptocurrency markets, thus enhancing the robustness and relevance of the analysis.

As a initial step, Let's see the descriptive statistics.

```
##       BTC                 ETH                 USDT          
##  Min.   :-0.371695   Min.   :-0.423472   Min.   :-5.121e-02  
##  1st Qu.:-0.016853   1st Qu.:-0.022792   1st Qu.:-6.519e-04  
##  Median : 0.001049   Median : 0.000437   Median :-1.174e-05  
##  Mean   : 0.002716   Mean   : 0.003105   Mean   : 1.121e-05  
##  3rd Qu.: 0.022311   3rd Qu.: 0.029543   3rd Qu.: 5.719e-04  
##  Max.   : 0.252472   Max.   : 0.409905   Max.   : 5.824e-02  
##       BNB                 XRP                X.GSPC          
##  Min.   :-0.591591   Min.   :-0.423341   Min.   :-0.1198406  
##  1st Qu.:-0.024183   1st Qu.:-0.026175   1st Qu.:-0.0045981  
##  Median : 0.001750   Median :-0.001076   Median : 0.0007578  
##  Mean   : 0.008528   Mean   : 0.003494   Mean   : 0.0004115  
##  3rd Qu.: 0.030270   3rd Qu.: 0.025231   3rd Qu.: 0.0066219  
##  Max.   : 2.300210   Max.   : 0.871500   Max.   : 0.0938277  
##     DX.Y.NYB              CL.F                GC.F          
##  Min.   :-0.0211669   Min.   :-3.059661   Min.   :-0.049787  
##  1st Qu.:-0.0024394   1st Qu.:-0.011868   1st Qu.:-0.003971  
##  Median : 0.0000000   Median : 0.002491   Median : 0.000468  
##  Mean   : 0.0000853   Mean   :-0.001523   Mean   : 0.000339  
##  3rd Qu.: 0.0024723   3rd Qu.: 0.014187   3rd Qu.: 0.004975  
##  Max.   : 0.0165245   Max.   : 0.376623   Max.   : 0.059477
```

The above summaries give us important understanding about the spread of data points. We observe that for all currencies/stocks, the minimum return is always negative. among the cyptocurrencies BNB has minimum return, while the highest is for BNB only. Let's look at some usual plots/charts related to cryptocurrencies.The plots show the dependence structure of the returns of the five mentioned cryptocurrencies, which indicates the use of copula to capture or model these dependency structures.

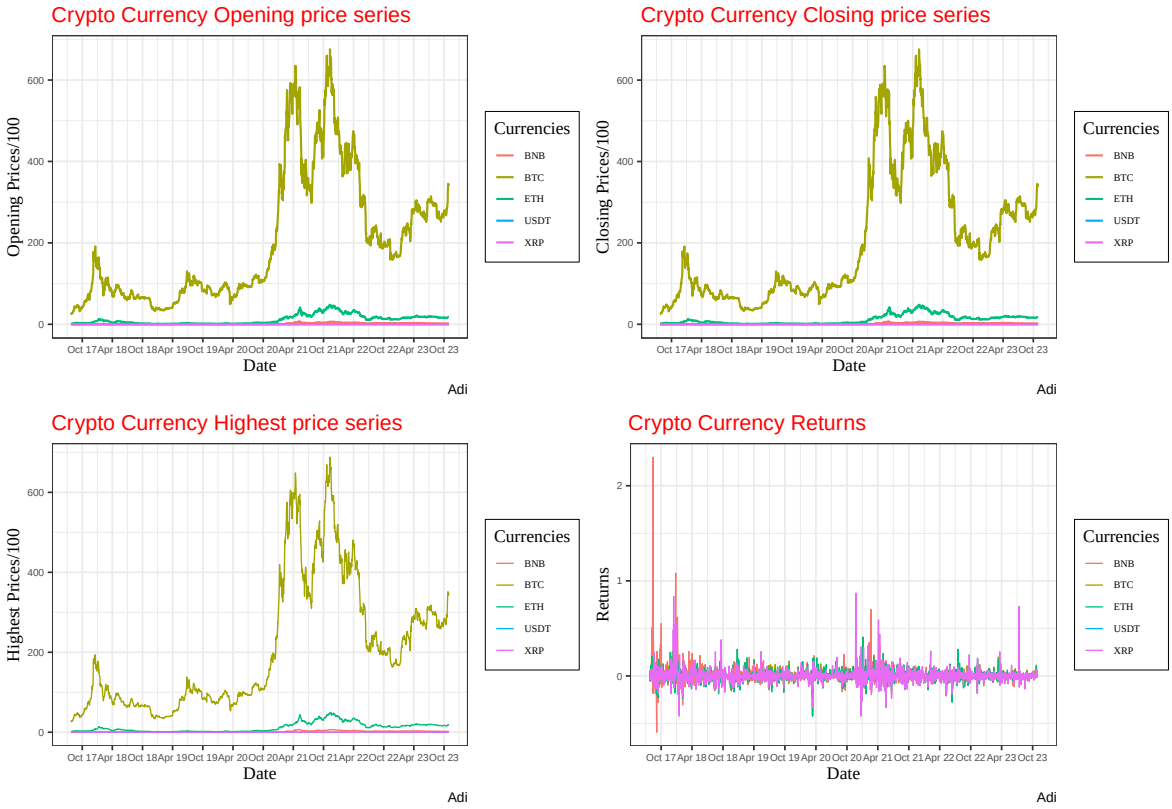

Figure 1: Plot of Opening, Closing, Highest and Returns different cryptocurrencies

We recognize the typical characteristics of this market, i.e. high volatility and sudden waves of exponential price increases. Daily log-returns of the five cryptocurrencies confirm their high volatility, a strong degree of comovement, and show the typical volatility clustering in common with other traditional financial assets. We observe volatility jumps not only during the first crypto bubble but also during the financial market crash caused by the COVID-19 pandemic and right after Biden's election at the end of 2020, confirming that crypto investors' reactions do not differ from the behavior of investors in traditional financial markets.

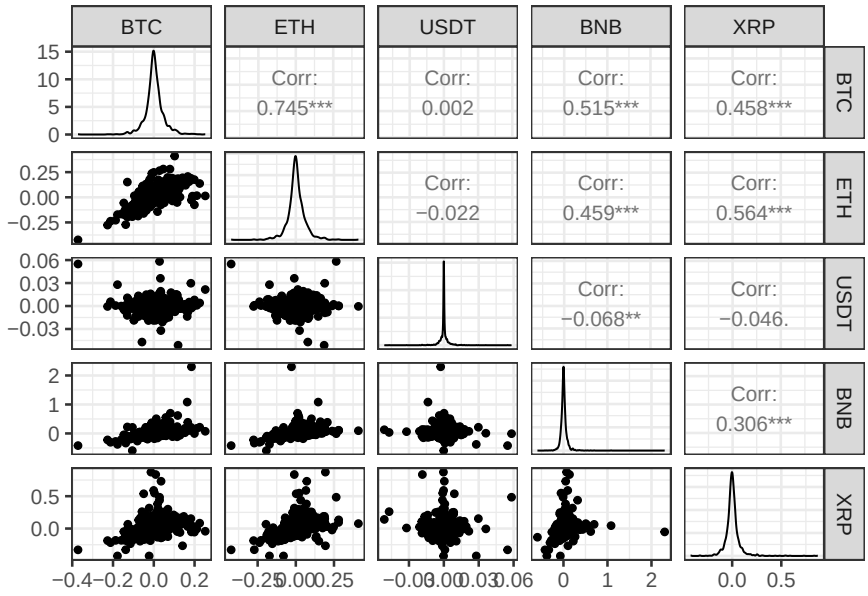

Figure 2: Paiwise Scatterplots of five cryptocurrency returns

Fiting The Proposed Model

We will fit the proposed model in R. But there is no built in package available for this. We will rather write our own function. We first recall the parameters of our model. Following the notation of original paper. The parameters of the model are $\boldsymbol{\theta} = (\beta_1, ..., \beta_K, \sigma_1, ..., \sigma_K, \pi, \Pi, \eta_1, ..., \eta_K)$. For Gaussian Copula, $\eta_k = (\Omega_k^{\Phi})$ and For t Copula, $\eta_k = (\Omega_k^{\Psi}, \nu_k)$, k = 1,2,. . . ,K. Where K is the number of states of hidden markov model.

To estimate the model parameters, we need to maximize the surrogate function w.r.t to θ -

$$Q(\theta|\theta^{(h)}) = \sum_{k=1}^{K} \gamma_1^{(h)}(k) log\ \pi_k + \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{K} \xi_i^{(h)}(j,k) log\ \pi_{k|j} + \sum_{t=1}^{T}\sum_{k=1}^{K} \gamma_t^{(h)}(k) log f_{Y_t}(y_t|x_t, S_t = k)$$

Where, $\gamma_t^{(h)}(k)$ and $\xi_i^{(h)}(j,k)$ are quantities depending upon *forward* and *backward* probabilities. Such quantities can be effectively find out using *Forward-Backward algorithm* and S_t denotes markov chain of the hidden markov model.

Parameter Initialization:

Before starting parameter estimation, we need to initialize parameter values i.e. the θ vector. Following Maruotti et al. (2021) and Merlo et al. (2022), for fixed $\boldsymbol{\tau}$ and K we initialize the EM algorithm by providing the initial states partition, $\{S_t^{(0)}\}_{t=1}^{T}$ according to a multinomial distribution with probabilities $1/K$. From the generated partition, the elements of $\Pi^{(0)}$ are computed as proportions of transition, while we obtain regression parameters $\boldsymbol{\beta^{(0)}}$ and $\boldsymbol{\sigma^{(0)}}$ by fitting univariate mean and median regressions on the observations within state k for the CQHMM and CEHMM, respectively. The state-dependent correlation matrices of the copula are set equal to the empirical correlation matrices computed on observations in the k-th state, while the initial value for the degrees of freedom of the t copula is

$\nu^{(0)} = \nu_k^{(0)} = 5$ for all $k = 1, \ldots, K$. Once we computed the ML estimates of the model parameters, to estimate the standard errors we employ the parametric bootstrap scheme of Visser et al. (2000).

Baum-Welch Algorithm:

Given the initial parameter values, we need to start finding the quantities $\gamma_t^{(h)}(k)$ and $\xi_i^{(h)}(j,k)$. This can be found using **Baum-Welch Algorithm**.

$$\gamma_i(t) = P(s_t = i|y_1, .., y_T, \theta^{(h)}) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{T} \alpha_j(t)\beta_j(t)}$$

$$\xi_{jk}(t) = P(s_t = k, s_{t-1} = j|y_1, ...y_T, \theta^{(h)}) = \frac{\alpha_k(t-1)m_{jk}\beta_k(t)g_k(y_t)}{\sum_{i=1}^{T}\sum_{j=1}^{T} \alpha_k(t-1)m_{jk}\beta_k(t)g_k(y_t)}$$

Where, $\alpha_j(t)$ and $\beta_j(t)$ are found using forward and backward algorithm. $m_{jk} = P(S_t = k|S_{t-1} = j)$ and $g_k(.)$ denote the emission distribution. i.e. distribution of y_t given $S_t = k$ and also $\boldsymbol{x_t}$. Details of this algorithm is available here https://mzaradzki.github.io/probabilistic-javascript/demos/hmm.html.

Results

We fit the proposed CQHMM using t copulas for values of K = 2,3. For ease of comparison between the two models, the copula function and K are selected for $\tau = \tau_j = 0.05/0.50/0.95, j = 1, ..., d$. Then, in the analysis, we have kept this choice fixed. In order to clearly identify high and low volatility market conditions, we use K = 2, which is supported by the parsimonious criteria for both CQHMM and CEHMM, together with a t copula. From a graphical perspective, the figures report the scatterplots and the marginal densities colored according to the estimated posterior probability of class membership, $\max_k\ \gamma_t(k)$.

First we present results for $K = 2$

$\tau_i = 0.05$, $\nu_k = 5$ and $K = 2$

State-1

	BTC	ETH	USDT	BNB	XRP
(Intercept)	-0.0846449	-0.0509983	-0.0360541	-0.1088325	-0.071126
S&P 500	-0.2300913	4.2215010	-0.3929791	5.3651761	12.955647
USDX	5.2632504	10.3068502	-1.2671049	13.8621074	36.921639
WTI Crude Oil	0.3213948	-3.1392648	0.2318785	-2.4034287	-9.119338
Gold	-4.5401410	-12.4363199	0.9301731	-15.8093942	-21.047584

	BTC	ETH	USDT	BNB	XRP
sigma1	0.0964965	0.1493678	0.0367004	0.1939656	0.3348341

State-2

	BTC	ETH	USDT	BNB	XRP
(Intercept)	-0.0665050	-0.0813797	-0.0049786	-0.0847997	-0.0925407
S&P 500	0.0981568	0.1761913	0.0399464	0.3496364	-0.1215364
USDX	2.0291683	1.0178805	-0.0078601	2.5190580	-0.3238101
WTI Crude Oil	-0.0203153	-0.0098503	-0.0008596	-0.0296758	-0.0308839
Gold	0.3002664	-0.0244694	0.0269448	0.5024830	0.1396120

	BTC	ETH	USDT	BNB	XRP
sigma2	0.0727067	0.0896047	0.0054907	0.0989487	0.1012585

$\tau_i = 0.50$**,** $\nu_k = 5$ **and** $K = 2$

State-1

	BTC	ETH	USDT	BNB	XRP
(Intercept)	-0.0086683	0.1585371	0.0042716	0.0389791	0.5300873
S&P 500	-1.9554851	-0.7244973	-0.0424364	-2.5683913	-12.2038059
USDX	-10.0145485	4.1004348	-6.9104478	-3.8905728	-15.4559146
WTI Crude Oil	2.9782645	8.8920892	4.0625801	-2.8125878	3.2437870
Gold	-5.8024912	-9.9691329	-6.0369696	7.6354559	39.3907995

	BTC	ETH	USDT	BNB	XRP
sigma1	0.0853625	0.2640831	0.0970255	0.111947	0.5935721

State-2

	BTC	ETH	USDT	BNB	XRP
(Intercept)	0.0009201	0.0002955	-0.0000186	0.0021080	-0.0009832
S&P 500	-0.1875282	-0.1912720	-0.0009105	-0.2027944	-0.2121419
USDX	0.0551657	-0.0494954	0.0015746	-0.0581235	0.0576136
WTI Crude Oil	0.0014352	0.0003008	-0.0005324	-0.0006772	-0.0005906
Gold	-0.0298612	-0.0203165	0.0019907	0.0378424	-0.0209166

	BTC	ETH	USDT	BNB	XRP
sigma2	0.0306217	0.0392125	0.0020242	0.0466735	0.043961

$\tau_i = 0.95$**,** $\nu_k = 5$ **and** $K = 2$

State-1

	BTC	ETH	USDT	BNB	XRP
(Intercept)	0.1725874	0.1860709	0.0327344	1.707701	0.5530456
S&P 500	12.6249052	8.8454218	-0.2505077	60.743735	6.4149188
USDX	-1.2169006	19.7350534	-5.5413971	-215.742978	8.3046943
WTI Crude Oil	-1.9108639	5.0792367	1.4675061	-9.141110	-1.3943211
Gold	0.2631298	-0.5855383	-6.9883477	-65.459024	59.3489333

	BTC	ETH	USDT	BNB	XRP
sigma1	0.2042794	0.2487529	0.0604927	1.817103	0.6645935

State-2

	BTC	ETH	USDT	BNB	XRP
(Intercept)	0.0805278	0.0962674	0.0053921	0.1122435	0.1033309
S&P 500	-0.7176846	-0.9401772	-0.0119915	-0.7270175	-0.4949979
USDX	-1.3591050	0.9543505	-0.0592745	-1.5589006	-0.0832296
WTI Crude Oil	0.0286328	0.0372808	0.0011606	0.0348811	0.0287689
Gold	-0.4822267	0.2483546	-0.0414702	-1.0526807	0.1249866

	BTC	ETH	USDT	BNB	XRP
sigma2	0.0806482	0.0977526	0.005931	0.1161326	0.1106707

The estimates of the state-specific parameters are gathered in the attached tables for the CQHMM, As regards the estimated scale parameters, σ_2 reflects stable periods, representing the so-called low-volatility state, meanwhile σ_1 contemplates rapid (positive and negative) peak and burst returns, which defines the second state as the high-volatility state. These results confirm the graphical analysis.

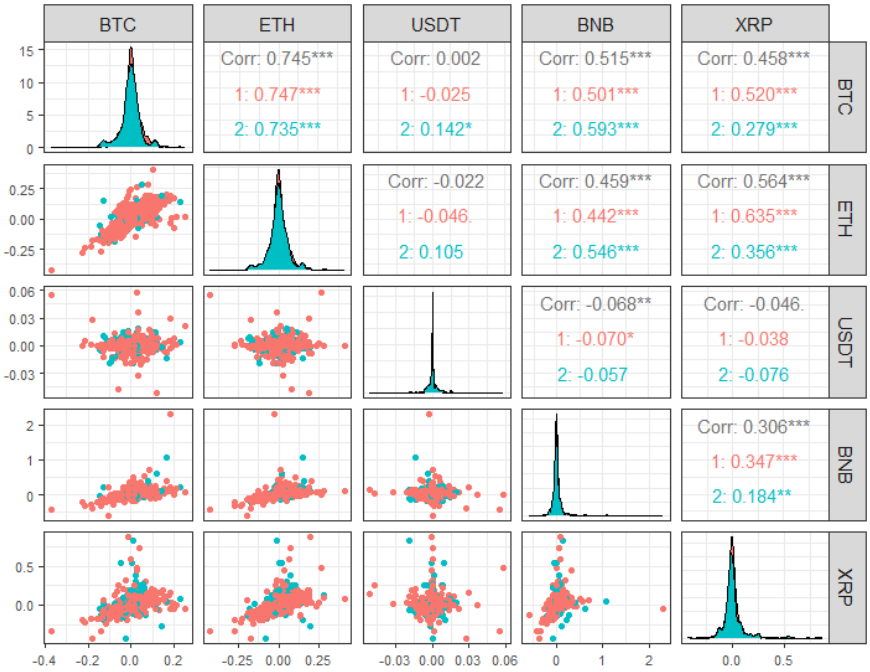

Figure 3: Scatterplot of State-Specific Crypto Returns for K = 2 and tau = 0.5

Taking the impact of covariates into account, we comment on the parameter estimates of the CQHMM. As could be expected, the state-specific intercepts are increasing somewhat with τ , with State 2 having lower absolute values than State 1 for all τ's. For $\tau = 0.50$ we observe that at low volatility periods S&P 500 is the only statistically significant asset, negatively influencing almost all the cryptocurrencies for both quantile and expectile models, which implies that during tranquil periods crypto assets can be considered as weak hedges, During high volatility periods we observe that S&P500, Gold and Crude Oil influence some cryptos, especially Bitcoin. In particular, for the CQHMM at $\tau = 0.05$, we observe that at high volatility periods Gold influences all the cryptos considered. Finally, at $\tau = 0.95$ for the CQHMM for both states considered we note strong influences of US dollar index and Gold. The results for $K = 3$ are shown below. But they are not that interesting as $K = 2$ case and also from model selection criterion (like - AIC,BIC) point of view also $K = 2$ is more interesting.

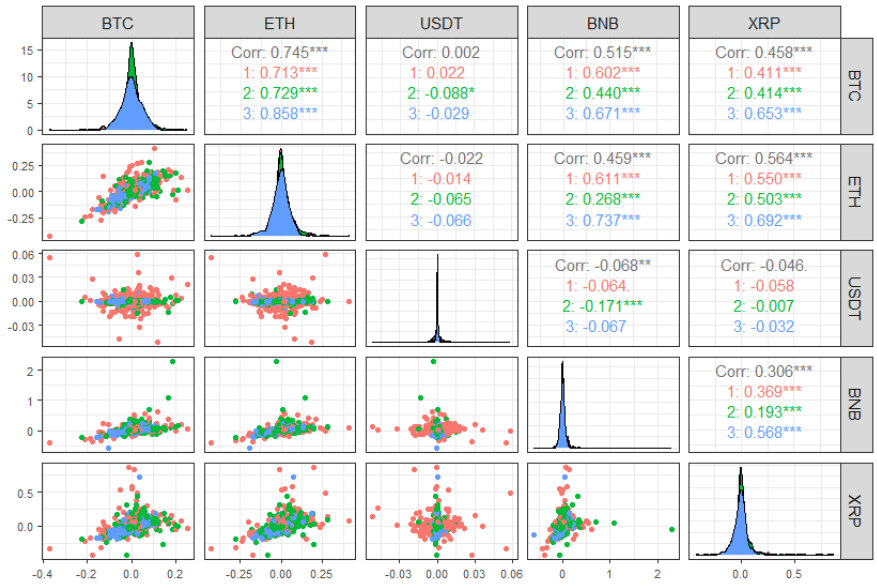

Figure 4: Scatterplot of State-Specific Crypto Returns for K = 3 and tau = 0.5

$\tau_i = 0.05$, $\nu_k = 5$ **and** $K = 3$

State-1

BTC	ETH	USDT	BNB	XRP
-0.0665050	-0.0813797	-0.0049117	-0.0848452	-0.0931602
0.0981568	0.1761913	0.0358050	0.3552558	-0.0825124
2.0291683	1.0178805	0.0086492	2.5275154	-0.2307904
-0.0203153	-0.0098503	-0.0006440	-0.0297113	-0.0310407
0.3002664	-0.0244694	0.0248006	0.5054907	0.2308490

	BTC	ETH	USDT	BNB	XRP
sigma1	0.0727067	0.0896047	0.0054324	0.098986	0.101767

State-2

BTC	ETH	USDT	BNB	XRP
-0.0859305	-0.0137011	-0.0360898	-0.0947809	-0.0621145
-0.2521259	2.4234452	-0.3899336	4.0901429	12.5756489
5.2317450	9.1670691	-1.2805400	20.7894684	30.8726725
0.1541987	-2.4792176	0.2314861	-2.1578552	-9.6246805
-4.2881094	-7.1637518	0.9162672	-9.7223289	-20.8062004

	BTC	ETH	USDT	BNB	XRP
sigma2	0.0966845	0.1046934	0.036733	0.1667933	0.32815

State-3

	BTC	ETH	USDT	BNB	XRP
(Intercept)	-0.0661196	-0.0568132	-0.0360898	-0.0384670	-0.0502038
S&P 500	-1.2002223	-0.9208480	-0.3899336	-0.7652794	-0.7535989
USDX	3.9516172	9.0850249	-1.2805400	4.4173353	7.0810976
WTI Crude Oil	0.4280673	0.0383404	0.2314861	-0.2783085	-0.0180251
Gold	-1.6113579	-2.0858266	0.9162672	-1.1869129	-1.5275868

	BTC	ETH	USDT	BNB	XRP
sigma3	0.0757316	0.0773146	0.036733	0.0671012	0.0720178

$\tau_i = 0.50$, $\nu_k = 5$ **and** $K = 3$

State-1

BTC	ETH	USDT	BNB	XRP
0.063561	0.0779725	0.0015886	0.2816032	0.0402373
7.297886	6.5143271	1.5441916	-23.6432966	6.3341326
-19.398519	42.6276484	-3.1561882	-439.0459182	29.7119658
-3.754430	4.3513656	-3.0179295	-12.6364850	3.1051597
19.617616	4.9257603	-1.0106403	178.8638577	-3.2067158

	BTC	ETH	USDT	BNB	XRP
sigma1	0.0727067	0.0896047	0.0054324	0.098986	0.101767

State-2

BTC	ETH	USDT	BNB	XRP
-0.0086685	0.1585371	0.0042716	0.0389791	0.5300873
-1.9554860	0.7244973	-0.0424364	-2.5683913	-12.2038059
-10.9145485	4.1004348	-6.9194478	-3.8905728	-15.4559146
2.9782645	8.8920892	4.0625801	-2.8125878	3.2437870
-5.8024912	-9.9691329	-6.0369693	7.6354559	39.3907995

	BTC	ETH	USDT	BNB	XRP
sigma2	0.0966845	0.1046934	0.036733	0.1667933	0.32815

State-3

	BTC	ETH	USDT	BNB	XRP
(Intercept)	0.0009174	0.0002577	-0.0000186	0.0020982	-0.0009832
S&P 500	-0.1876269	-0.2352813	-0.0009105	-0.2028221	-0.2121419
USDX	0.0568084	-0.1602186	0.0015746	-0.0589008	0.0576136
WTI Crude Oil	0.0014366	0.0004490	-0.0005324	-0.0006833	-0.0005906
Gold	-0.0295625	-0.0336622	0.0019907	0.3676164	-0.0209166

	BTC	ETH	USDT	BNB	XRP
sigma3	0.0757316	0.0773146	0.036733	0.0671012	0.0720173

$\tau_i = 0.95$, $\nu_k = 5$ **and** $K = 3$

State-1

BTC	ETH	USDT	BNB	XRP
0.1817063	0.1860709	0.0414017	1.750372	0.5738683
13.6480949	8.8454218	0.5552804	64.123778	8.5040808
-0.2663764	19.7350534	-4.1253673	-209.905450	14.9392246
-2.5664559	5.0792367	0.5591093	-10.235225	-3.5718990
-1.0085077	-0.5855383	-6.3094919	-72.255621	60.5738749

	BTC	ETH	USDT	BNB	XRP
sigma1	0.2182792	0.2487529	0.054312	1.864498	0.6870796

State-2

BTC	ETH	USDT	BNB	XRP
0.0803419	0.0962674	0.0053910	0.1122435	0.1033309
-0.7620062	-0.9401772	-0.0119752	-0.7270175	-0.4949979
-1.2300351	0.9543505	0.0591747	-1.5589006	-0.0832296
0.0289898	0.0372808	0.0011599	0.0348811	0.0287689
-0.4477699	0.2483546	-0.0415804	-1.0526807	0.1249866

	BTC	ETH	USDT	BNB	XRP
sigma2	0.0804852	0.0977526	0.0059299	0.1161326	0.1106707

State-3

	BTC	ETH	USDT	BNB	XRP
(Intercept)	0.1842176	0.1860709	0.0419249	1.762073	0.5749594
S&P 500	13.7218200	8.8454522	0.6039198	64.474049	8.6135497
USDX	0.2007593	19.7350534	-4.0398922	-207.735485	15.2868639
WTI Crude Oil	-2.4553320	5.0792367	0.5042762	-9.726902	-3.6860008
Gold	-1.4837502	-0.5855383	-6.2685144	-74.465965	60.6380600

	BTC	ETH	USDT	BNB	XRP
sigma3	0.2202129	0.2487529	0.0541788	1.873412	0.688337

Observations

1. The relationship among these two different markets is rather complex, and it is more pronounced in the tails of the returns distributions.

2. The dependence within the crypto market varies over time according to the market conditions.

The considered timespan extends from July, 25 2017 to October, 31 2023, including numerous crises that have impacted cross-market integration patterns, such as the crypto price bubbles of early 2018, the COVID-19 pandemic, Biden's election at the USA presidency in November 2020 and the Russian invasion of Ukraine at the beginning of 2022, which have caused unprecedented levels of uncertainty and risk.

Theory Extension

Quantile and Expectile Copula Based Hidden Semi-Markov Model

This novel approach introduces the Copula Quantile Hidden Semi Markov Model (CQHSMM) and the Copula Expectile Hidden Semi Markov Model (CEHSMM) to capture state-dependent multivariate distributions. Emphasizing flexibility, the models incorporate copulas with dynamic parameters, allowing for accurate representation of complex dependence structures. The models employ Gaussian and t copulas, with sojourn durations modeled through a semi-Markovian framework.

Let $\{S_t\}_{t=1}^T$ represent a finite-state hidden semi-Markov chain over a discrete state space $S = \{1, ..., K\}$. The latent process $\{S_t\}_{t=1}^T$ is constructed as follows. A homogeneous hidden Markov chain with K states models transitions between different states with initial probabilities, $\pi_k = Pr(S_1 = k)$, and transition probabilities $\pi_{jk} = Pr(S_{t+1} = k|S_t = j, S_{t+1} \neq j)$, with $\sum_{k=1}^K \pi_{jk} = 1$, $\pi_{jk} \geq 0$, for every $k = 1, ..., K$ and $\pi_{jj} = 0$. In short, we collect initial and transition probabilities in the K-dimensional vector π and in the $K \times K$ matrix Q, respectively. Because the unobserved process is semi-Markovian, only transitions between states are governed by the transition probabilities. Still, the duration of a stay in a state is modeled by a separate sojourn duration (SD). Let $d_k(u)$ be the SD, i.e., the probability that $\{S_t\}_{t=1}^T$ spends u consecutive time steps in the k-th state. U_k corresponds to the maximum sojourn time in the k-th state. Let $U = (U_1, ..., U_K)$ be the K-dimensional vector collecting all state-specific maximum sojourn times.

HSMMs offer flexibility as the SD is directly specified by the researcher and can be estimated from the observed data. The SD can be chosen from various parametric distributions, such as shifted-Poisson or shifted-negative binomial. Parametric distributions might lack flexibility, so semi- and nonparametric data-driven approaches can be adopted for more flexibility and better accommodation of complex distributional shapes.

To build the proposed model, let $\boldsymbol{Y_t} = (Y_{t,1}, ..., Y_{t,d})$ be a continuous observable d-variate response variable, and $\boldsymbol{x_t} = (1, x_{t,2}, ..., x_{t,p})$ be a p-dimensional vector of covariates. The process $\{Y_t\}_{t=1}^T$ represents the state-dependent process of the HSMM and, conditional on hidden states, fulfills the independence property.

$$f_Y(y_t|x_t, y_1, ..., y_{t-1}, S_1 = s_1, ..., S_t = s_t) = f_Y(y_t|x_t, S_t = s_t)$$

Let $F_{Y_{t,j}}(y_{t,j}|x_t, S_t = k)$, $j = 1, ..., d$, be the distribution functions of the marginals, the state-dependent multivariate distribution of Y_t given covariates and $S_t = k$.

$$F_{Y_t}(y_t|x_t, S_t = k) = C(F_{Y_{t,1}}(y_{t,1}|x_t, S_t = k), ..., F_{Y_{t,d}}(y_{t,d}|x_t, S_t = k); \eta_k)$$

By Sklar's Theorem, the joint density can be expressed as:

$$f_{Y_t}(y_t|x_t, S_t = k) = \Pi_{j=1}^d \ f_{Y_{t,j}}(y_{t,j}|x_t, S_t = k)c(u_1, ..., u_d; \eta_k) \qquad ...(1)$$

Here, $u_j = F_{Y_{t,j}}(y_{t,j}|x_t, S_t = k)$ and $c(.; \eta_k) = \dfrac{\partial^d C(.; \eta_k)}{\partial F_1 ... \partial F_d}$ ....(2)

If a particular form of the density function is used, then

$$f_{Y_{t,j}}(y_{t,j}|x_t, S_t = k) = B_{l,\tau_j}(\sigma_{j,k})exp[-\omega_{l,\tau_j}(\frac{y_{t,j} - \mu_{t,j,k}}{\sigma_{j,k}})]$$

where $\mu_{t,j,k}$ is the Location Parameter, $\sigma_{j,k} > 0$ is the Scale Parameter, $\omega_{l,\tau_j}(.)$ is the Kernel function related to the quantile/expectile loss function, and $B_{l,\tau_j}(\sigma_{j,k})$ is the Normalizing constant.

Now, $\mu_{t,j,k}(\tau_j) = x_t B_{j,k}(\tau_j);\ j = 1, ..., d.$ The above equations define the newly proposed Copula Quantile Hidden SeminMarkov Model when $l = 1$ and Copula Expectile Hidden Semi Markov Model when $l = 2$. The distribution functions of Gaussian and t copulas are written as -

$$C^G(\boldsymbol{u}, \Omega^\Phi) = \Phi_d(\Phi^{-1}(u_1), ..., \Phi^{-1}(u_d); \Omega^\Phi)$$

$$C^t(\boldsymbol{u}, \Omega^\Psi, \nu) = \Psi_d(\Psi^{-1}(u_1; \nu), ..., \Psi^{-1}(u_d; \nu); \Omega^\Psi, \nu)$$

where Φ_d and Ψ_d denote the joint distribution functions of d-variate normal and t distribution with correlation matrices Ω^Φ and Ω^Ψ respectively. Φ^{-1} and Ψ^{-1} are the inverse distribution functions of univariate standard distributions. We impose $\nu > 2$ on the degrees of freedom of t copula.

Appendix - Codes

The code provided below offers a step-by-step guide to model estimation and analysis. This implementation bridges the theoretical concepts outlined in the paper with tangible application, demonstrating how these advanced models can be utilized to unravel the complex dynamics of cryptocurrency markets in real-world scenarios.

CQHMM Model

Paremeter Initialization:

```
y_dummy <- our_data[-1,2:6]
x_dummy <- our_data[-nrow(our_data),7:10]
our_data <- data.frame(t = 1:nrow(y_dummy),y_dummy,x_dummy)
tau_fixed <- 0.95

T <- nrow(our_data); K <- 2
hidden_chain <- sample(1:K,size = T,replace = T)
transition_prob <- as.matrix(markovchainFit(hidden_chain)$estimate[1:K,])

beta0_1 <- vector(mode = 'list',length = K)
sigma0_1 <- vector(mode = "list",length = K)

#CQHMM
for(i in 1:K){
  m1 = lm(BTC ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
          data = our_data[hidden_chain == i,])
  m2 = lm(ETH ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
          data = our_data[hidden_chain == i,])
  m3 = lm(USDT ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
          data = our_data[hidden_chain == i,])
  m4 = lm(BNB ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
          data = our_data[hidden_chain == i,])
  m5 = lm(XRP ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
          data = our_data[hidden_chain == i,])
  beta0_1[[i]] <- cbind(m1$coefficients,m2$coefficients,
                        m3$coefficients,m4$coefficients,
                        m5$coefficients)
  sigma0_1[[i]] <- c(sum(m1$residuals^2)/m1$df.residual,
                     sum(m2$residuals^2)/m2$df.residual,
                     sum(m3$residuals^2)/m3$df.residual,
                     sum(m4$residuals^2)/m4$df.residual,
                     sum(m5$residuals^2)/m5$df.residual)

}

#Initial Correlation Matrix
cormat0 <- vector(mode = 'list',length = K)
for(i in 1:K){
```

```
  cormat0[[i]] <- cor(our_data[hidden_chain == i,2:6])
}
```

Emission Distribution Using t-Copula:

```
G <- function(y,x,betas,sigmas){
  dens_val <- NULL;k <- length(sigmas)
  for(i in 1:k){
   #myCop <- tCopula(param = 5,dim = 5,dispstr = "un", correlation = cormat0[[i]])
   myCop <- tCopula(df = 5,dim = 5,param = cormat0[[i]][lower.tri(cormat0[[i]])],
                    dispstr = "un")
   myMvd <- mvdc(copula=myCop, margins=c("ALD","ALD","ALD","ALD","ALD"),
                 paramMargins=list(list(mu = sum(c(1,x)*betas[[i]][,1]),
                                        sigma = sigmas[[i]][1],p = tau_fixed),
                                   list(mu = sum(c(1,x)*betas[[i]][,2]),
                                        sigma = sigmas[[i]][2],p = tau_fixed),
                                   list(mu = sum(c(1,x)*betas[[i]][,3]),
                                        sigma = sigmas[[i]][3],p = tau_fixed),
                                   list(mu = sum(c(1,x)*betas[[i]][,4]),
                                        sigma = sigmas[[i]][4],p = tau_fixed),
                                   list(mu = sum(c(1,x)*betas[[i]][,5]),
                                        sigma = sigmas[[i]][5],p = tau_fixed)))
    dens_val <- c(dens_val,dMvdc(y,myMvd))
  }
  return(dens_val)
}
```

Baum-Welch Algorithm:

```
# forward-backward algorithm with parameter estimation
forward_backward_em_algorithm <- function(y,x,Q,G,m,betas,sigmas){

  T <- nrow(y)  # Number of time points
  K <- nrow(Q)    # Number of states
  prox <- rep(0.0001,K)

  # Initialization
  alpha <- matrix(0,nrow = T, ncol = K)
  beta <- matrix(0,nrow = T, ncol = K)

  # E-step (Forward-Backward)
  alpha[1, ] <- m * G(as.vector(as.matrix(y[1,])),
                      as.vector(as.matrix(x[1,])),betas,sigmas) + 0.0001
  for (t in 2:T) {
    g <- G(as.vector(as.matrix(y[t,])),as.vector(as.matrix(x[t,])),betas,sigmas)
    alpha[t, ] <- apply(Q * crossprod(t(alpha[t - 1, ]),g + prox), 2, sum) + 0.0001
    alpha[t, ] <- alpha[t, ] / sum(alpha[t, ])
```

```
  }

  beta[T, ] <- rep(1, K)
  for (t in (T - 1):1) {
    g <- G(as.vector(as.matrix(y[t+1,])),as.vector(as.matrix(x[t+1,])),betas,sigmas)
    beta[t, ] <- apply(Q * crossprod(t(beta[t + 1, ]), (g + prox)), 1, sum) + 0.0001
    beta[t, ] <- beta[t, ] / sum(beta[t, ])
  }

  # Combine forward and backward probabilities to get smoothed probabilities
  gamma_h <- alpha * beta
  gamma_h <- gamma_h / rowSums(gamma_h)

  epsilon <- vector(mode='list', length = length(y))
  epsilon[[1]] = t(replicate(K,gamma_h[1,]))
  for(t in 2:T){
    g <- G(as.vector(as.matrix(y[t,])),as.vector(as.matrix(x[t,])),betas,sigmas)
    epsilon[[t]] = t(t(Q)*alpha[t-1,]*(g + prox)*beta[t,]) + 0.0001
  }

  epsilon <- lapply(epsilon,FUN = function(mat){mat/sum(mat)})
  hidden_var <- apply(gamma_h,1,FUN = function(arr){which.max(arr)})

  return(list(alpha = alpha, beta = beta, gamma = gamma_h,
              epsilon = epsilon,state = hidden_var))
}

y = our_data[,2:6]
x = our_data[,7:10]
Q = transition_prob
m = rep(1,K)/K
results = forward_backward_em_algorithm(y,x,Q,G,m,beta0_1,sigma0_1)
```

Iterations to estimate parameters:

```
for(epoch in 1:20){

  new_gamma_h <- results$gamma
  new_m <- results$gamma[1,]
  new_hidden_var <- results$state
  new_Q <- Reduce(`+`,results$epsilon)
  new_Q <- new_Q/rowSums(new_Q)
  K <- nrow(new_Q)

  new_betas <- vector(mode = 'list',length = K)
  new_sigmas <- vector(mode = "list",length = K)

  for(i in 1:K){
    m1 = rq(BTC ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
```

```
            tau = tau_fixed,weights = new_gamma_h[,i])
    m2 = rq(ETH ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
            tau = tau_fixed,weights = new_gamma_h[,i])
    m3 = rq(USDT ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
            tau = tau_fixed,weights = new_gamma_h[,i])
    m4 = rq(BNB ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
            tau = tau_fixed,weights = new_gamma_h[,i])
    m5 = rq(XRP ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
            tau = tau_fixed,weights = new_gamma_h[,i])
    new_betas[[i]] <- cbind(m1$coefficients,m2$coefficients,
                            m3$coefficients,
                            m4$coefficients,
                            m5$coefficients)
    new_sigmas[[i]] <- c(mean(abs(m1$residuals)),mean(abs(m2$residuals)),
                         mean(abs(m3$residuals)),
                         mean(abs(m4$residuals)),mean(abs(m5$residuals)))
  }


  #Initial Correlation Matrix
  new_cormat <- vector(mode = 'list',length = K)
  for(i in 1:K){
    new_cormat[[i]] <- cor(our_data[new_hidden_var == i,2:6])
  }

  new_G <- function(y,x,betas,sigmas){
    dens_val <- NULL;k <- length(sigmas)
    for(i in 1:k){
      #myCop <- tCopula(param = 5,dim = 5,dispstr = "un", correlation = cormat0[[i]])
      myCop <- tCopula(df = 5,dim = 5,param = cormat0[[i]][lower.tri(new_cormat[[i]])],
                       dispstr = "un")
      myMvd <- mvdc(copula=myCop, margins=c("ALD","ALD","ALD","ALD","ALD"),
                    paramMargins=list(list(mu = sum(c(1,x)*betas[[i]][,1]),
                                           sigma = sigmas[[i]][1],p = tau_fixed),
                                      list(mu = sum(c(1,x)*betas[[i]][,2]),
                                           sigma = sigmas[[i]][2],p = tau_fixed),
                                      list(mu = sum(c(1,x)*betas[[i]][,3]),
                                           sigma = sigmas[[i]][3],p = tau_fixed),
                                      list(mu = sum(c(1,x)*betas[[i]][,4]),
                                           sigma = sigmas[[i]][4],p = tau_fixed),
                                      list(mu = sum(c(1,x)*betas[[i]][,5]),
                                           sigma = sigmas[[i]][5],p = tau_fixed)))
      dens_val <- c(dens_val,dMvdc(y,myMvd))
    }
    return(dens_val)
  }

  results = forward_backward_em_algorithm(y,x,new_Q,new_G,new_m,
                                          new_betas,new_sigmas)
}
```

Bootstrap to estimate std error:

```
G_sample <- function(y,x,betas,sigmas){

  myCop <- tCopula(df = 5,dim = 5,param = cormat0[[i]][lower.tri(cormat0[[i]])],
                   dispstr = "un")
  myMvd <- mvdc(copula=myCop, margins=c("ALD","ALD","ALD","ALD","ALD"),
                paramMargins=list(list(mu = sum(c(1,x)*betas[,1]),
                                       sigma = sigmas[1],p = tau_fixed),
                                  list(mu = sum(c(1,x)*betas[,2]),
                                       sigma = sigmas[2],p = tau_fixed),
                                  list(mu = sum(c(1,x)*betas[,3]),
                                       sigma = sigmas[3],p = tau_fixed),
                                  list(mu = sum(c(1,x)*betas[,4]),
                                       sigma = sigmas[4],p = tau_fixed),
                                  list(mu = sum(c(1,x)*betas[,5]),
                                       sigma = sigmas[5],p = tau_fixed)))
  return(rMvdc(y,myMvd))
}
```

Using the above function, we have drawn sample from the estimation distribution repeatedly. For each such sample we have estimated model parameters and then have calculated the standard error of the model.

CEHMM Model

Paremeter Initialization:

```
y_dummy <- our_data[-1,2:6]
x_dummy <- our_data[-nrow(our_data),7:10]
our_data <- data.frame(t = 1:nrow(y_dummy),y_dummy,x_dummy)
tau_fixed <- 0.95

T <- nrow(our_data); K <- 2
hidden_chain <- sample(1:K,size = T,replace = T)
transition_prob <- as.matrix(markovchainFit(hidden_chain)$estimate[1:K,])

#CEHMM
beta0_2 <- vector(mode = 'list',length = K)
sigma0_2 <- vector(mode = "list",length = K)
for(i in 1:K){
  m1 = rq(BTC ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
          data = our_data[hidden_chain == i,],tau = 0.5)
  m2 = rq(ETH ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
          data = our_data[hidden_chain == i,],tau = 0.5)
  m3 = rq(USDT ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
          data = our_data[hidden_chain == i,],tau = 0.5)
  m4 = rq(BNB ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
```

```
          data = our_data[hidden_chain == i,],tau = 0.5)
  m5 = rq(XRP ~ X.GSPC + DX.Y.NYB + CL.F + GC.F,
          data = our_data[hidden_chain == i,],tau = 0.5)
  beta0_2[[i]] <- cbind(m1$coefficients,m2$coefficients,
                        m3$coefficients,m4$coefficients,
                        m5$coefficients)
  sigma0_2[[i]] <- c(mean(abs(m1$residuals)),
                     mean(abs(m2$residuals)),mean(abs(m3$residuals)),
                     mean(abs(m4$residuals)),mean(abs(m5$residuals)))
}

#Initial Correlation Matrix
cormat0 <- vector(mode = 'list',length = K)
for(i in 1:K){
  cormat0[[i]] <- cor(our_data[hidden_chain == i,2:6])
}
```

After initializing the parameters we will use t-Copula and Baum-Welch Algorithm to update the parameters. Then we iterate to get final estimates.

Iterations to estimate parameters:

```
for(epoch in 1:20){

  new_gamma_h <- results$gamma
  new_m <- results$gamma[1,]
  new_hidden_var <- results$state
  new_Q <- Reduce(`+`,results$epsilon)
  new_Q <- new_Q/rowSums(new_Q)
  K <- nrow(new_Q)

  new_betas <- vector(mode = 'list',length = K)
  new_sigmas <- vector(mode = "list",length = K)

  for(i in 1:K){
    m1 = rq(BTC ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
            tau = tau_fixed,weights = new_gamma_h[,i])
    m2 = rq(ETH ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
            tau = tau_fixed,weights = new_gamma_h[,i])
    m3 = rq(USDT ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
            tau = tau_fixed,weights = new_gamma_h[,i])
    m4 = rq(BNB ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
            tau = tau_fixed,weights = new_gamma_h[,i])
    m5 = rq(XRP ~ X.GSPC + DX.Y.NYB + CL.F + GC.F ,data = our_data,
            tau = tau_fixed,weights = new_gamma_h[,i])
    new_betas[[i]] <- cbind(m1$coefficients,m2$coefficients,
                            m3$coefficients,m4$coefficients,
                            m5$coefficients)
    new_sigmas[[i]] <- c(mean(abs(m1$residuals)),mean(abs(m2$residuals)),
```

```
                          mean(abs(m3$residuals)),
                          mean(abs(m4$residuals)),mean(abs(m5$residuals)))
  }


  #Initial Correlation Matrix
  new_cormat <- vector(mode = 'list',length = K)
  for(i in 1:K){
    new_cormat[[i]] <- cor(our_data[new_hidden_var == i,2:6])
  }

  new_G <- function(y,x,betas,sigmas){
    dens_val <- NULL;k <- length(sigmas)
    for(i in 1:k){
      #myCop <- tCopula(param = 5,dim = 5,dispstr = "un", correlation = cormat0[[i]])
      myCop <- tCopula(df = 5,dim = 5,param = cormat0[[i]][lower.tri(new_cormat[[i]])],
                       dispstr = "un")
      myMvd <- mvdc(copula=myCop, margins=c("ALD","ALD","ALD","ALD","ALD"),
                    paramMargins=list(list(mu = sum(c(1,x)*betas[[i]][,1]),
                                           sigma = sigmas[[i]][1],p = tau_fixed),
                                      list(mu = sum(c(1,x)*betas[[i]][,2]),
                                           sigma = sigmas[[i]][2],p = tau_fixed),
                                      list(mu = sum(c(1,x)*betas[[i]][,3]),
                                           sigma = sigmas[[i]][3],p = tau_fixed),
                                      list(mu = sum(c(1,x)*betas[[i]][,4]),
                                           sigma = sigmas[[i]][4],p = tau_fixed),
                                      list(mu = sum(c(1,x)*betas[[i]][,5]),
                                           sigma = sigmas[[i]][5],p = tau_fixed)))
      dens_val <- c(dens_val,dMvdc(y,myMvd))
    }
    return(dens_val)
  }

  results = forward_backward_em_algorithm(y,x,new_Q,new_G,new_m,
                                          new_betas,new_sigmas)
}
```

References

1. Quantile and expectile copula-based hidden Markov regression models for the analysis of the cryptocurrency market
2. Hidden Markov Models
3. crypto2 R package
4. Quantile models for multivariate time series