



HARVARD UNIVERSITY
The Graduate School of Arts and Sciences

ADVANCED SCIENTIFIC COMPUTING: NUMERICAL METHODS
AM205

Do crypto currencies move in parallel manner?

Students:

Yihang Yan
Tomas Gudmundsson
Nathaniel Stein

Professor:

Chris Rycroft

December 15, 2017

Table of contents

1	Introduction	3
2	Principal Components Analysis	3
2.1	(a) Motivation	3
2.2	(b) Mathematics of Principal Components	4
2.2.1	Singular Value Decomposition (SVD)	4
2.2.2	Principal component analysis (PCA) by SVD	5
2.2.3	Finding the components	6
2.2.4	Contribution of an observation to a component	6
3	Jacobi eigenvalue algorithm	7
4	Implementation	12
4.1	(a) Data	12
4.2	(b) Computing eigenvectors and corresponding eigenvalues	13
4.3	(c) Sorting Eigenpairs and Explained Variance	14
4.4	(d) Component Loadings	15
5	Do Cryptocurrencies Move in a Parallel Manner?	19
5.1	(a) Existing Literature	20
5.2	(b) Proportion test	20
5.3	(c) Implications	21
5.4	(d) Inter-temporal consistency of covariance matrix	22

1 Introduction

This paper examines the statistical distribution of the price return patterns of cryptocurrencies through the lens of principal component analysis (PCA). The rise of cryptocurrencies is the most intensely debated paradigm shift in financial markets this past decade, with opinions ranging from JPMorgan CEO Jamie Dimon calling them a fraud to Bill Gates claiming Bitcoin is better than currency. To add to the unfolding drama, the Chicago Board Options Exchange (CBOE) debuted the trading of Bitcoin futures on December 10, 2017. A discussion on the merits and future of these innovations is beyond the scope of this paper.

Instead, we will assume cryptocurrencies are not going away anytime soon and focus our analysis on other issues pertinent to investors and market-makers in cryptocurrencies. First, we will use PCA to assess whether the price returns of cryptocurrencies move in a parallel manner. Second, we sample different time frames to determine whether there have been any meaningful structural shifts in the price patterns of cryptocurrencies. Finally, we highlight the implications of our results for portfolio construction, risk management, and trading.

This paper is structured as follows. The following section explains the motivation for applying PCA on financial asset returns and the mathematical theory behind the method before applying it to cryptocurrency price returns. Section 3 will use the results of these PCA findings to address the question of whether cryptocurrencies move in a parallel manner, and Section 4 will contextualize the significance of these findings by assessing whether the price patterns of cryptocurrencies have demonstrated significant inter-temporal changes. Our final section (Section 5) summarizes the conclusions of our analysis. Although PCA is routinely applied to studies of traditional asset classes and incorporated in various econometric models, to the best of this papers authors knowledge, it has not been applied in as rigorous of a fashion to cryptocurrency price returns.

2 Principal Components Analysis

2.1 (a) Motivation

Before delving into the specifics of this papers implementation of PCA, let us motivate the benefits for studying asset returns with PCA. Principal components analysis (PCA) is a multivariate technique that analyzes a data in which observations are described by several inter-correlated quantitative dependent variables. This technique extracts the important information from the data to represent it as a set of new orthogonal variables called principal components, or factors.

Factor models explaining stock returns and correlations have been very popular in finance. Unlike traditional factor models, the factors created by PCA do not usually have a clear economic interpretation. Rather, the components constructed in PCA are built to have special statistical characteristics whereby each component:

- Aims to account for as much of variation in the data as possible.

- Is uncorrelated with every other component, i.e., the components are orthogonal to each other.

In short, for the purposes of this paper, PCA enables the identification of the underlying statistical factors that cause the comovement in cryptocurrency returns. These findings can be used to quantify the relative importance of each cryptocurrency in explaining the movement of the cryptocurrency market as a whole.

The first principal component is the component that best explains variation in the underlying data, i.e., the greatest amount of variation, and is of particular importance to this study. In finance, risk is frequently broken down into two categories: *systematic* (i.e., market) risk that can be mitigated through a diversified portfolio and *idiosyncratic* risk that is specific to each individual asset and cannot be diversified away. In applications of PCA on asset returns, the first principal component is generally accepted as a representation of the overall return of the assets, arguably representing the return to investors for taking on the systematic risk of those assets (Shukla and Trzcinka, 1990). In other words, if all the assets shared the same idiosyncratic risks, the first principal component could be conceptualized as the equal weighted market index (Shukla and Trzcinka, 1990). Thus, for assets that are fairly correlated with one another, i.e., a large proportion of their comovement is accounted for by the general fortunes (or misfortunes) in their market, one would expect the first component to account for a relatively large proportion of variance and to have similar loadings on the variables.

Mathematically, PCA depends upon the eigendecomposition of positive semi-definite matrices and upon the singular value decomposition (SVD) of rectangular matrices.

2.2 (b) Mathematics of Principal Components

2.2.1 Singular Value Decomposition (SVD)

Any real symmetric $m \times m$ matrix A has a spectral decomposition of the form,

$$A = U\Delta U^T \quad (1)$$

where U is an orthonormal matrix (matrix of orthogonal unit vectors: $U^T U = I$ or $\sum_k U_{ki} U_{kj} = \delta_{ij}$) and Δ is a diagonal matrix. The columns of U are the eigenvectors of matrix A and the diagonal elements of Δ are the eigenvalues. If A is positive-definite, the eigenvalues will all be positive. Multiplying with U , equation 1 can be re-written to,

$$AU = U\Delta U^T U = UA \quad (2)$$

This can be written as a normal eigenvalue equation by defining the i th column of U as u_i and the eigenvalues as $\lambda_i = \Delta_{ii}$:

$$Au_i = \lambda_i u_i \quad (3)$$

Let's look at more general case. An unsymmetrical ($n \times m$) matrix, where $n \geq m$ B has the decomposition,

$$X = U\Delta V^T \quad (4)$$

where U is a $n \times m$ matrix with orthonormal columns ($U^T U = I$), while V is a $m \times m$ orthonormal matrix ($V^T V = I$), and Δ is a $m \times m$ diagonal matrix with positive or zero elements, called the singular values.

From B we can construct two positive-definite symmetric matrices, BB^T and $B^T B$, each of which we can decompose

$$BB^T = U\Delta V^T V \Delta U^T = U\Delta^2 U^T \quad (5)$$

$$B^T B = V\Delta^2 V^T \quad (6)$$

We can now show that BB^T which is $n \times n$ and $B^T B$ which is $m \times m$ will share m eigenvalues and the remaining $n - m$ eigenvalues of BB^T will be zero.

Using the decomposition above, we can identify the eigenvectors and eigenvalues for BB^T as the columns of U and the squared diagonal elements of Δ , respectively. Denoting one such eigenvector by u and the diagonal element by γ , we have:

$$B^T B u = \gamma^2 u \quad (7)$$

$$BB^T B u = \gamma^2 B u \quad (8)$$

This means that we have an eigenvector $u = Bv$ and eigenvalue γ^2 for BB^T as well, since:

$$(BB^T) B u = \gamma^2 B u \quad (9)$$

We have now shown that $B^T B$ and BB^T share m eigenvalues.

In order to prove that the remaining $n - m$ eigenvalues of BB^T is zero. We need to consider an eigenvector for BB^T , u_\perp : $BB^T u_\perp = \beta_\perp u_\perp$ which is orthogonal to the m eigenvectors u_i already determined, i.e. $U^T u_\perp = 0$. Using the decomposition $BB^T = U\Delta^2 U^T$, we immediately see that the eigenvalues β_\perp must all be zero,

$$BB^T u_\perp = U\Delta^2 U^T u_\perp = 0 u_\perp \quad (10)$$

2.2.2 Principal component analysis (PCA) by SVD

We denote the matrix of eigenvectors sorted according to eigenvalue by \hat{U} and we can then PCA transformation of the data as $Y = \hat{U}^T X$. The eigenvectors are called the principal components. By selecting the first d rows of Y , we can project the data from n down to d dimensions.

We decompose X using SVD, i.e.

$$X = U\Delta V^T \quad (11)$$

and find that we can write the covariance matrix as

$$C = \frac{1}{n}XX^T = \frac{1}{n}U\Delta^2U^T \quad (12)$$

Following from the fact that SVD routine order the singular values in descending order we know that, if $n < m$, the first n columns in U corresponds to the sorted eigenvalues of C and if $m \geq n$, the first m corresponds to the sorted non-zero eigenvalues of C . The transformed data can thus be written as:

$$Y = \hat{U}^T X = \hat{U}^T U \Delta V^T \quad (13)$$

where $\hat{U}^T U$ is a simple $n \times m$ matrix which is one on the diagonal and zero everywhere else. So we can write the transformed data in terms of the SVD decomposition of X .

2.2.3 Finding the components

In PCA, the components are obtained from the singular value decomposition of the dataset X . Specifically, with $X = U\Delta V^T$ (equation 1), the matrix of factor scores, denoted F is obtained as

$$F = U\Delta \quad (14)$$

The matrix V gives the coefficients of the linear combinations used to compute the factors scores. This matrix can also be interpreted as a projection matrix because multiplying X by V gives the values of the projections of the observations on the principal components. This can be shown as:

$$F = U\Delta = U\Delta VV^T = XV \quad (15)$$

The components can be represented geometrically by the rotation of the original axes. Each of these components will be linear combinations of the observed variables we have in our data, and will be orthogonal to each other. That is, each component is independent of each other, and variation in one is unrelated to variation in another.

2.2.4 Contribution of an observation to a component

The eigenvalue associated to a component is equal to the sum of the squared factor scores for this component. Therefore, the importance of an observation for a component can be obtained by the ratio of the squared factor score of this observation by the eigenvalue associated with that component. This ratio is called the contribution of the observation to the component. Formally, the contribution of observation i to component l is denoted $ctr_{i,l}$, it is obtained as:

$$ctr_{i,l} = \frac{f_{i,l}^2}{\sum f_{i,l}^2} = \frac{f_{i,l}^2}{\lambda_l} \quad (16)$$

where λ_l is the eigenvalue of the l th component. The value of a contribution is between 0 and 1 and, for a given component, the sum of the contributions of all observations is equal to 1. The larger the value of the contribution, the more the observation contributes to the component.

3 Jacobi eigenvalue algorithm

In order to find eigenvalues and eigenvectors of the covariance matrix, A , we use the Jacobi method. The method works by finding the largest non-diagonal element at location (i, j) and makes it zero by doing a plane rotation on rows and columns i and j . The Jacobi method is quite efficient with quadratic convergence and can be parallelized easily.

The rotation for matrix A works as follows:

$$A' = P_{i,j,\theta}^T \cdot A \cdot P_{i,j,\theta} \quad (17)$$

where $P_{i,j,\theta} = P(i, j, \theta)$ is a Givens rotation matrix with the form:

$$P(i, j, \theta) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \quad (18)$$

This matrix has ones on the diagonal except where $c = \cos(\theta)$ at locations (i,i) and (j,j) . All other elements are 0 except $s = \sin(\theta)$ at locations (i,j) and (j,i) .

When the Givens rotation matrix, $P_{i,j,\theta}$, is multiplied with another matrix, A , as $P \cdot A$ it simulates a clockwise rotation in the plane by an angle θ in order to make the element at location (i, j) zero and only affects rows and columns i and j in the process.

In order to figure out the values of c, s and θ we will simulate a matrix multiplication with 2×2 matrices that contain the relevant information. We will represent the matrices as follows:

$$P_{i,j,\theta} = \begin{bmatrix} c_{ii} & s_{ij} \\ -s_{ji} & c_{jj} \end{bmatrix}, A = \begin{bmatrix} A_{ii} & A_{ij} \\ A_{ji} & A_{jj} \end{bmatrix}, A' = \begin{bmatrix} A'_{ii} & A'_{ij} \\ A'_{ji} & A'_{jj} \end{bmatrix} \quad (19)$$

where the notation A_{ij} denotes element at location (i, j) in matrix A .

Now we will find c, s and θ so that the matrix multiplication zeros the largest element. In order for P to satisfy eq. 17 we expand the matrix multiplication as follows with $c_{ii} = c_{jj} = c$ and $s_{ji} = s_{ij} = s$:

$$\begin{aligned}
\begin{bmatrix} A'_{ii} & A'_{ij} \\ A'_{ji} & A'_{jj} \end{bmatrix} &= \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \cdot \begin{bmatrix} A_{ii} & A_{ij} \\ A_{ji} & A_{jj} \end{bmatrix} \cdot \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \\
&= \begin{bmatrix} c \cdot A_{ii} - s \cdot A_{ji} & c \cdot A_{ij} - s \cdot A_{jj} \\ s \cdot A_{ii} + c \cdot A_{ji} & s \cdot A_{ij} + c \cdot A_{jj} \end{bmatrix} \cdot \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \\
&= \begin{bmatrix} c^2 \cdot A_{ii} - cs \cdot A_{ji} - cs \cdot A_{ij} + s^2 \cdot A_{jj} & cs \cdot A_{ii} - s^2 \cdot A_{ji} + c^2 \cdot A_{ij} - cs \cdot A_{jj} \\ cs \cdot A_{ii} + c^2 \cdot A_{ji} - s^2 \cdot A_{ij} - cs \cdot A_{jj} & s^2 \cdot A_{ii} + cs \cdot A_{ji} + cs \cdot A_{ij} + c^2 \cdot A_{jj} \end{bmatrix} \\
&= \begin{bmatrix} c^2 \cdot A_{ii} - 2cs \cdot A_{ij} + s^2 \cdot A_{jj} & (c^2 - s^2) \cdot A_{ij} + cs \cdot (A_{ii} - A_{jj}) \\ (c^2 - s^2) \cdot A_{ij} - cs \cdot (A_{ii} - A_{jj}) & c^2 \cdot A_{jj} + 2cs \cdot A_{ij} + s^2 \cdot A_{ii} \end{bmatrix}
\end{aligned} \tag{20}$$

where the last elements are simplified because $A_{ij} = A_{ji}$.

In order to make the non diagonal element in this matrix as 0 we will examine the non diagonal equation as follows:

$$A'_{ij} = (c^2 - s^2) \cdot A_{ij} + cs(A_{ii} - A_{jj}) = 0 \tag{21}$$

Hence it follows that:

$$\frac{c^2 - s^2}{cs} = \frac{A_{jj} - A_{ii}}{A_{ij}} \tag{22}$$

and we can define the rotation angle as follows:

$$\theta = \cot(2\phi) = \frac{c^2 - s^2}{2cs} = \frac{A_{jj} - A_{ii}}{2A_{ij}} \tag{23}$$

and by letting $t = s/c$ we can rewrite the equation above as:

$$2cs\theta = c^2 - s^2 \Leftrightarrow t^2 + 2t\theta - 1 = 0 \tag{24}$$

which has the solutions:

$$t = \begin{cases} -\theta + \sqrt{\theta^2 + 1} \\ -(\theta + \sqrt{\theta^2 + 1}) \end{cases} \tag{25}$$

These solutions can be written more succinctly as

$$t = -\theta + \sqrt{\theta^2 + 1} = \frac{(-\theta + \sqrt{\theta^2 + 1})(-\theta + \sqrt{\theta^2 + 1})}{\theta + \sqrt{\theta^2 + 1}} = \frac{-\theta^2 + \theta^2 + 1}{\theta + \sqrt{\theta^2 + 1}} = \frac{1}{\theta + \sqrt{\theta^2 + 1}} \tag{26}$$

$$t = -\theta - \sqrt{\theta^2 + 1} = \frac{(-\theta - \sqrt{\theta^2 + 1})(\theta - \sqrt{\theta^2 + 1})}{\theta - \sqrt{\theta^2 + 1}} = \frac{-\theta^2 + \theta^2 + 1}{\theta - \sqrt{\theta^2 + 1}} = \frac{-1}{-\theta + \sqrt{\theta^2 + 1}} \tag{27}$$

We want to rotate the matrix by the angle which corresponds to the smaller root of this equation and generally we can write the smaller root as:

$$t = \frac{\text{sign}(\theta)}{|\theta| + \sqrt{1 + \theta^2}} \quad (28)$$

and since $t = s/c$ we now have:

$$c = \frac{1}{\sqrt{t^2 + 1}}, \quad s = t \cdot c \quad (29)$$

Now we know how to set these variables for the rotations to work and we need to look at three scenarios to update the matrix when a rotation is performed:

- i) Set value at location (i,j) as 0
- ii) Change diagonal values at locations (i,i) and (j,j)
- iii) Change values on rows and columns i and j except (i,i) and (j,j)

The following diagram shows which values are modified during the rotation:

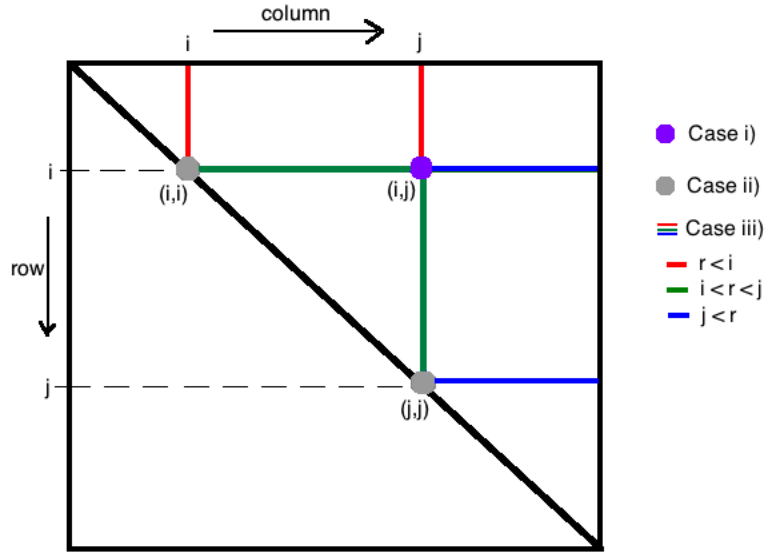


Figure 1: Visual representation of how the Jacobi plane rotations work

We define a tolerance $tol = 10^{-9}$ and perform rotations until the largest non-diagonal element is less than the tolerance. Since the matrix is symmetric we will perform rotations on the upper triangle of the matrix until it has converged.

For scenario i) we simply set the value of A'_{ij} as 0. However, for scenario ii) we will look at the top left and bottom right elements in eq. 20 to gather equations to set the diagonal elements A'_{ii} and A'_{jj} . We have:

$$A'_{ii} = c^2 \cdot A_{ii} - 2cs \cdot A_{ij} + s^2 \cdot A_{jj} \quad (30)$$

From eq. 21 (because $A'_{ij} = 0$) we can isolate A_{jj} as

$$A_{jj} = A_{ii} - A_{ij} \frac{s^2 - c^2}{cs} \quad (31)$$

and since $c^2 + s^2 = 1$ we simplify eq 14. as:

$$\begin{aligned} A'_{ii} &= c^2 \cdot A_{ii} - 2cs \cdot A_{ij} + s^2 \cdot A_{jj} \\ &= c^2 \cdot A_{ii} - 2cs \cdot A_{ij} + s^2 \left(A_{ii} - A_{ij} \frac{s^2 - c^2}{cs} \right) \\ &= (c^2 + s^2) \cdot A_{ii} - s \left(2c + \frac{s^2 - c^2}{c} \right) A_{ij} \\ &= (c^2 + s^2) \cdot A_{ii} - \frac{s}{c} (2c^2 + s^2 - c^2) A_{ij} \\ &= A_{ii} - \frac{s}{c} (c^2 + s^2) A_{ij} \\ &= A_{ii} - t \cdot A_{ij} \end{aligned} \quad (32)$$

Similarly we have:

$$A'_{jj} = A_{jj} + t \cdot A_{ij} \quad (33)$$

For scenario iii) we can look at top of eq 4. and note that if we consider an element A_{rj} when we perform rotation around A_{ij} that only the last two matrices will change the result since the first matrix changes rows i and j and does not have effect on row r. The last matrix changes columns i and j and therefore changes the resulting matrix. Multiplying through these matrices gives us the equations:

$$\begin{cases} A'_{ri} = cA_{ri} - sA_{rj} \\ A'_{rj} = cA_{ri} + sA_{rj} \end{cases} \quad (34)$$

Lets look at A'_{ri} which can be represented as:

$$\begin{aligned}
A'_{ri} &= cA_{ri} - sA_{rj} \\
&= \left(1 - \frac{(1-c)(1+c)}{1+c}\right) A_{ri} - sA_{rj} \\
&= \left(1 - \frac{1-c^2}{1+c}\right) A_{ri} - sA_{rj} \\
&= \left(1 - \frac{s^2}{1+c}\right) A_{ri} - sA_{rj} \\
&= A_{ri} - s \left(A_{rj} + \frac{s}{1+c} A_{ri}\right) \\
&= A_{ri} - s (A_{rj} + \tau A_{ri})
\end{aligned} \tag{35}$$

where

$$\tau = \frac{s}{1+c} \tag{36}$$

which has less roundoff error than eq. 34.

By performing the Jacobi rotations until the matrix converges we will end up with the eigenvalues at the diagonals. To get the eigenvectors for the eigenvalues we will initialise an identity matrix, P , and each time we update the matrix A we will update P accordingly. The eigenvector for each eigenvalue will be the same column it belongs to.

Similarly we have

$$A'_{rj} = A_{rj} + s (A_{ri} - \tau A_{rj}) \tag{37}$$

To summarise we set values of elements in rows r and l and columns r and l as follows:

$$\text{i) } A_{ij} = 0$$

$$\text{ii) } \begin{cases} A'_{ii} = A_{ii} - t \cdot A_{ij} \\ A'_{jj} = A_{jj} + t \cdot A_{ij} \end{cases}$$

$$\text{iii) } \begin{cases} A'_{ri} = A_{ri} - s (A_{rj} + \tau A_{ri}) \\ A'_{rj} = A_{rj} + s (A_{ri} - \tau A_{rj}) \end{cases}$$

where

$$s = t \cdot c, \quad t = \frac{\text{sign}(\theta)}{|\theta| + \sqrt{1 + \theta^2}}, \quad \tau = \frac{s}{1+c}, \quad \theta = \frac{A_{ii} - A_{jj}}{2A_{ij}}$$

4 Implementation

4.1 (a) Data

Our raw dataset consists of the closing prices for the top seven cryptocurrencies in terms of market capitalization (as listed on CoinMarketCaps' historical tables) for the period between August 7, 2015 to November 7, 2017. The start of the period is the first day for which a closing price on all seven of the selected cryptocurrencies was available because some, like Ripple, did not come into existence until much after the first cryptocurrencies, like Bitcoin. Using this dataset, we first created a time-series matrix containing rolling month-to-month changes in the closing prices for these cryptocurrencies, i.e., rolling monthly returns. For example, the return for Bitcoin (btc) on September 6, 2015 is equal to the percentage change in its closing price from August 7, 2015 (279.58) to September 6, 2015 (239.84): $239.84/279.58 - 1 = -14.21\%$.

We then standardized the returns for each cryptocurrency by centering around that particular currency's mean return and scaling by its standard deviation to achieve unit variance. The correlation matrix of this standardized dataset serves as the input for PCA. The correlation matrix is typically used instead of the covariance matrix. However, the eigendecomposition of a covariance matrix formed on standardized data yields the same results as an eigendecomposition on a correlation matrix (whether done on standardized or unstandardized data), since the correlation matrix can be understood as the normalized covariance matrix. When using a covariance matrix on unstandardized financial time series data, it should be pointed out that the resulting PCA could lead to the first principal component being dominated by the asset with the largest variance. This is especially impactful when the variances of the individual assets are significantly different from one another.

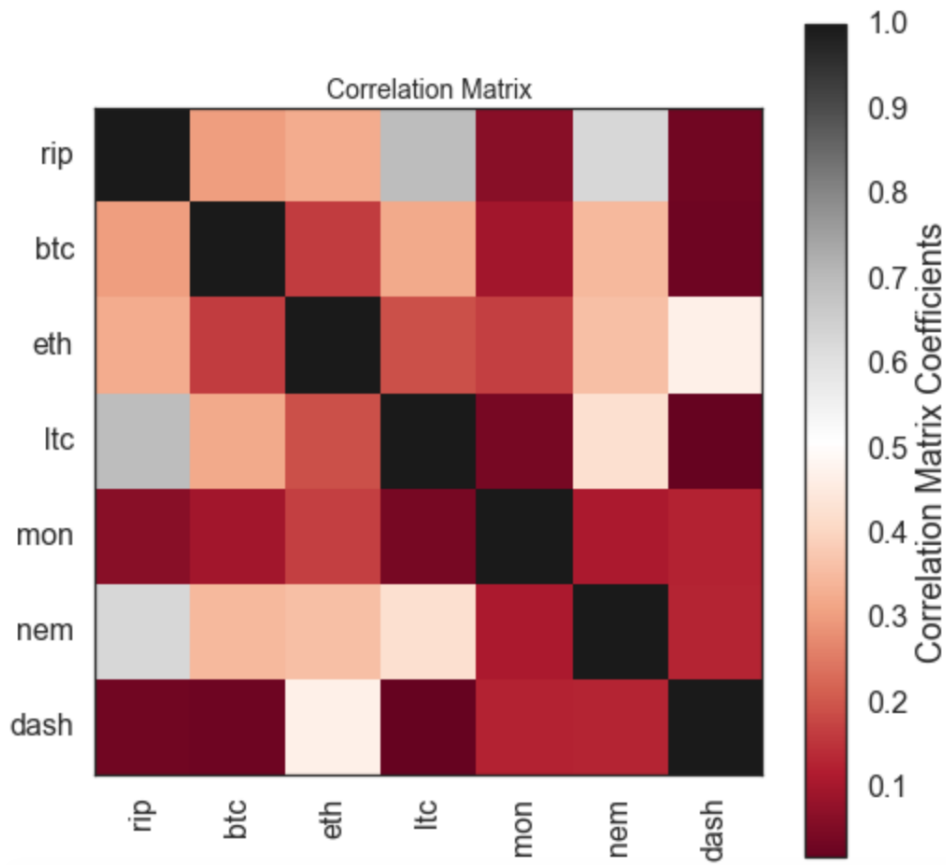


Figure 2: The correlation matrix of the crypto currencies

One of the basic assumptions behind using PCA is that the variables being examined have some kind of a linear relationship with each other; else, it would be unlikely that they share any meaningful common components. To assess the strength of the variables linear relationship and, consequently, suitability for PCA, one can use the correlation coefficients between the pairs of the variables (Hair, 2010). We conclude that the correlation between cryptocurrency pairs is strong enough to at least warrant further exploration.

4.2 (b) Computing eigenvectors and corresponding eigenvalues

We will use the Jacobi method to find eigenvalues and eigenvectors of the correlation matrix since this method is a fairly robust way to extract all of the eigenvalues and eigenvectors of a symmetric matrix. The method is based on a series of rotations, called Givens rotations, which are chosen to eliminate off-diagonal elements while preserving the eigenvalues. The algorithm finds the maximum non-diagonal element at index (i, j) by performing a scan over the matrix, zeros it out and then performs a plane rotation that affects elements in rows and columns i and j . During

each modification the algorithm also modifies a transformation matrix in the same way in order to keep track of the eigenvectors. This algorithm has a quadratic convergence and finishes when all non-diagonal elements are less than a chosen tolerance value. The eigenvalues are located at the diagonal of the input matrix and the eigenvectors are the columns of the transformation matrix. Further details of Jacobi method are covered in Section 3.

We can check if the eigenvector-eigenvalue calculation is correct by using the equation:

$$M_{cov}u_i = \lambda_i u_i$$

where

$$\begin{aligned} M_{cov} &= \text{covariance matrix,} \\ u_i &= \text{ith column of eigenvector matrix,} \\ \lambda_i &= \text{eigenvalue associated with } u_i \end{aligned}$$

and by testing it this way and against built in svd packages we have found that it gives the same results.

4.3 (c) Sorting Eigenpairs and Explained Variance

In order to decide which eigenvector(s) can be dropped without losing too much information for the construction of lower-dimensional subspace, we need to inspect the corresponding eigenvalues: The eigenvectors with the lowest corresponding eigenvalues bear the least information about the distribution of the data; those are the ones that can be dropped. In order to do so, the common approach is to rank the eigenvalues from highest to lowest in order to choose the top k eigenvectors. The first principal component is required to have the largest possible variance. The second component is computed under the constraint of being orthogonal to the first component and to have the largest possible inertia. The other components are computed likewise.

Principal Component	Eigenvalue	Eigenvector
1	2.654	array([0.517, 0.334, 0.358, 0.455, 0.137, 0.487, 0.174])
2	1.376	array([0.237, 0.15 , -0.508, 0.309, -0.327, 0.063, -0.676])
3	0.941	array([-0.105, 0.305, -0.204, -0.065, 0.88 , -0.037, -0.274])
4	0.769	array([-0.286, 0.867, 0.007, -0.226, -0.311, -0.044, 0.13])
5	0.564	array([-0.015, -0.095, 0.228, -0.611, -0.056, 0.64 , -0.39])
6	0.464	array([-0.013, -0.064, -0.712, -0.087, 0.027, 0.472, 0.508])
7	0.239	array([0.764, 0.091, -0.118, -0.511, 0.012, -0.351, 0.103])

Table 1: Eigenvalue and Eigenvector for each principal component

After sorting the eigenpairs, the next question is how many principal components are we going to choose for our new feature subspace? A useful measure is the proportion of variance, which can be

calculated from the eigenvalues. The explained variance tells us how much information (variance) can be attributed to each of the principal components.

Table 2 displays the results of a principal-component analysis of the cryptocurrencies daily returns: The eigenvalues, proportion of variance, and cumulative variance for each component. The single-strongest factor only explains 37.9% of the variation of crypto-currency returns. Moreover, each subsequent factor is providing only slowly declining additional information content, so that at least 5 factors are needed in order to account for 90% of the variation from these 7 crypto-currencies.

Principal Component	Eigenvalue	Proportion of Variance	Cumulative Variance
1	2.654	37.915%	37.915%
2	1.376	19.635%	57.55%
3	0.941	13.422%	70.972%
4	0.769	10.967%	81.939%
5	0.564	8.040%	89.979%
6	0.464	6.613%	96.592%
7	0.239	3.408%	100%

Table 2: Explained variance and cumulative variance for each principal component

?????? ADD (Comparison with XLE)

4.4 (d) Component Loadings

In multivariate space, the correlation between the principal component and the original variables (cryptocurrency price returns) is called the loading (or weight) of that component on the original variable. Based on loadings, we can tell how much of the variation in a variable is explained by the component.

$$\text{Loadings} = \text{Orthonormal Eigenvectors} * \sqrt{\text{Absolute Eigenvalues}}$$

Principal component loading diagram is shown below:

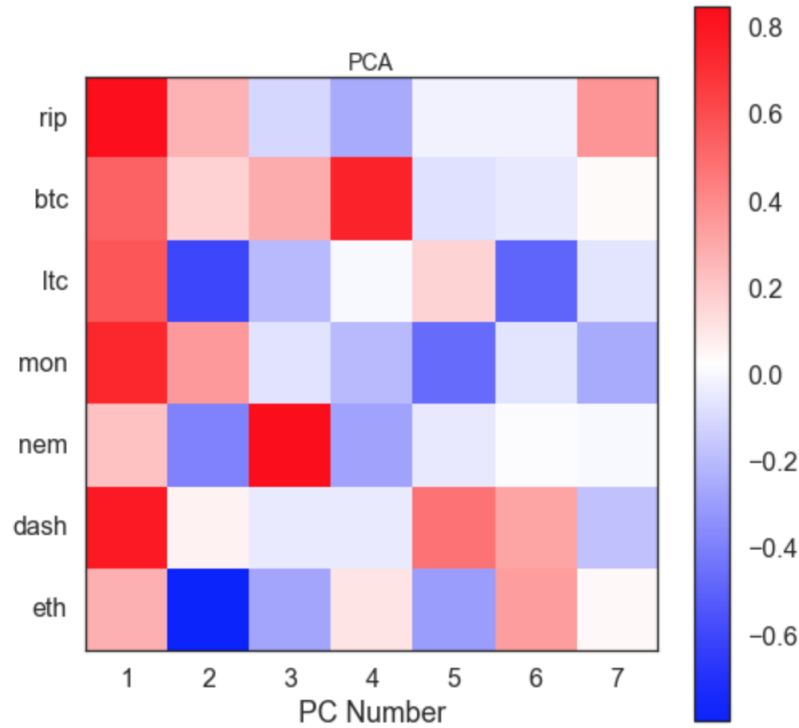


Figure 3: PCA matrix for 7 crypto currencies

Principal Component	Ripple	Bitcoin	Litecoin	Monero	Nem	Dash	Ethereum
1	0.844	0.544	0.583	0.742	0.224	0.793	0.284
2	0.278	0.176	-0.595	0.363	-0.383	0.074	-0.793
3	-0.102	0.295	-0.198	-0.063	0.853	-0.036	-0.266
4	-0.250	0.760	0.006	-0.198	-0.272	-0.039	0.114
5	-0.011	-0.071	0.171	-0.459	-0.042	0.481	-0.293
6	-0.009	-0.043	-0.485	-0.059	0.018	0.321	0.346
7	0.373	0.044	-0.058	-0.250	0.006	-0.171	0.051

Table 3: Component Loadings

The first principal component is strongly correlated (loading score ≥ 0.7) with three out of seven currencies. The first principal component increases with increasing Ripple, Monero, and Dash scores. This suggests that these three currencies are likely to vary together (positively correlated). If one increases, then the remaining ones tend to as well. Furthermore, we see that the correlation of first principal component with those six currencies are quite similar.

We also notice that the second principal component (negatively) correlates most strongly with Ethereum. In fact, we could state that based on the correlations of -0.793, this principal component is primarily a measure of Ethereum.

We construct the bi-plot of relative weights of each cryptocurrency in the first two PC components (PC-1 and PC-1) arising from the previous analysis:

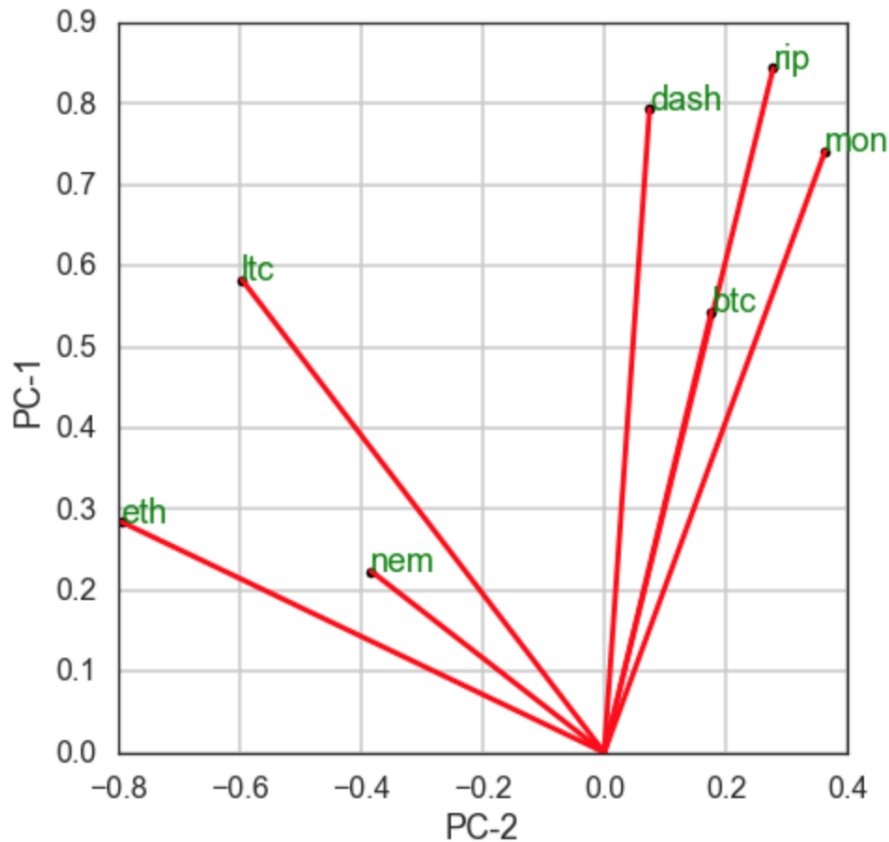


Figure 4: Bi-plot of relative weights for each crypto currency in the first two PC components

This diagram also demonstrate the distinct movement between Ethereum and the rest of cryptocurrencies. All seven variables have positive values in the PC1 axis, while Litecoin, Nem, and Ethereum are negative in PC2's and Ripple, Bitcoin, Monero and Dash is positive. Since all the variables are positive in PC1, those which constrain the system the most are Ripple and (then) Dash and Monero (in PC1 axis). The PC2, which has much smaller variance, contrasts Ethereum from everything else.

The cosine of the angles between vectors is equal to the correlation between those variables. Hence vectors pointing in the same direction are perfectly correlated, and those at right angles are uncorrelated. As we can see, Ripple and Bitcoin are high correlation, while Ethereum has extreme low correlation with Bitcoin, Monero, and Ripple. This conclusion is also supported by the correlation matrix.

In order to further confirm the weak correlation for each pair of currencies (cryptocurrency time-series), we make use of two distinct tools, namely, one-factor linear regression (hence its R^2 metric) and Kendall's rank correlation metric of τ . Specifically, Kendall's τ is calculated based on concordant and discordant pairs. It's insensitive to error and its P values are more accurate with smaller sample sizes. Correlation analyses can be used to test for associations in hypothesis testing. The null hypothesis is that there is no association between the variables under study. Thus, the purpose is to investigate the possible association in the underlying variables.

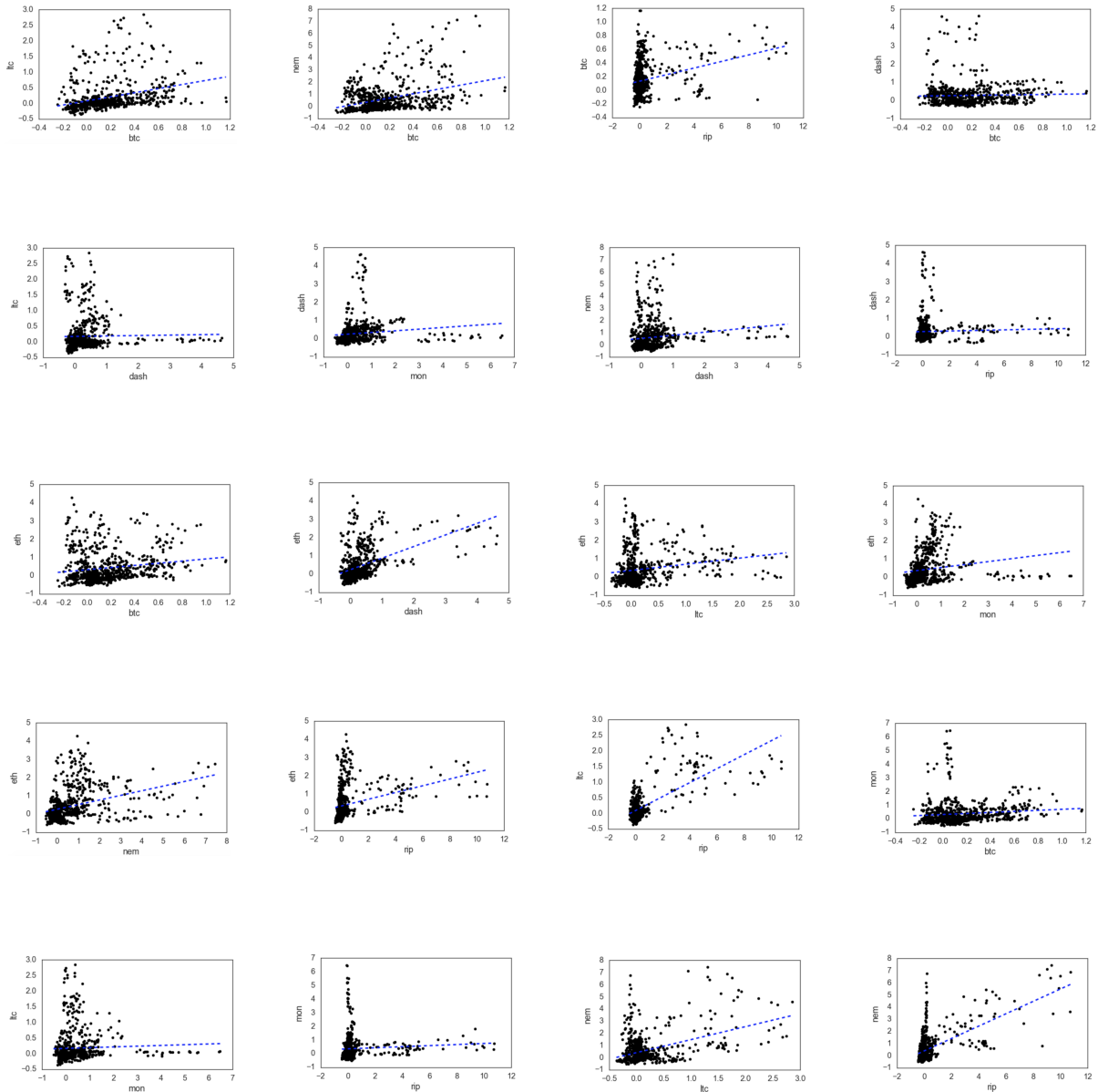


Figure 5: Yihang:

Currencies Pair	r2 - p value	tau - p value
btc - ltc	0.322 - $1.259e^{-20}$	0.364 - $4.667e^{-53}$
btc - nem	0.349 - $4.211e^{-24}$	0.193 - $4.066e^{-16}$
btc - rip	0.305 - $1.631e^{-18}$	0.146 - $6.484e^{-10}$
dash - btc	0.031 - 0.382	0.006 - 0.797
dash - ltc	0.018 - 0.608	0.082 - 0.001
dash - mon	0.128 - 0.0001	0.314 - $5.295e^{-40}$
dash - nem	0.133 - 0.0001	0.237 - $1.761e^{-23}$
dash - rip	0.037 - 0.298	0.077 - 0.001
eth - btc	0.164 - $3.196e^{-6}$	0.155 - $5.608e^{-11}$
eth - dash	0.470 - $6.202e^{-45}$	0.368 - $2.346e^{-54}$
eth - ltc	0.192 - $5.180e^{-8}$	0.198 - $7.079e^{-17}$
eth - mon	0.171 - $1.169e^{-6}$	0.317 - $7.959e^{-41}$
eth - nem	0.361 - $8.235e^{-26}$	0.312 - $1.714e^{-39}$
eth - rip	0.325 - $4.846e^{-21}$	0.321 - $1.164e^{-41}$
ltc - rip	0.694 - $4.391e^{-115}$	0.198 - $5.890e^{-17}$
mon - btc	0.102 - 0.004	0.159 - $1.980e^{-11}$
mon - ltc	0.044 - 0.220	0.133 - $1.840e^{-8}$
mon - rip	0.066 - 0.062	0.238 - $8.575e^{-24}$
nem - ltc	0.428 - $1.1262e^{-36}$	0.121 - $3.055e^{-7}$
nem - mon	0.114 - 0.001	0.286 - $1.637e^{-33}$
nem - rip	0.630 - $4.111e^{-89}$	0.233 - $8.714e^{-23}$

Table 4: R^2 metric and Kendall’s rank correlation metric of τ for each pair of cryptocurrencies

As we can see, indeed, a low degree in linear relationship is confirmed by $R^2 \leq 0.9$ at $\tau \leq 0.75$ for all of them.

The purpose of this section was to elucidate how PCA can be conducted on a financial time series like the universe of cryptocurrencies that are the focus of this paper. We have shown what the eigenvectors, components and loadings look like. Yet, we have not provided any point of reference in which to contextualize these results. This will be done in the next section, where we aim to answer the question of whether cryptocurrencies move in a parallel manner. To answer such a question, one needs to define what return patterns would justify being considered ”parallel.”. To do this, we will employ statistical tests done in other PCA studies on financial time series data and provide a point of reference to help compare and contrast the results of PCA on cryptocurrencies against another asset class.

5 Do Cryptocurrencies Move in a Parallel Manner?

On November 23, 2017, Fortune magazine ran the story titled ”Ethereum Price Hits New High as Billionaire Predicts 25% Surge In the Next Month.” Several articles like this appear daily for

seemingly each cryptocurrency, and each touts the purported benefits of its subject coin’s features over the rest. This all alludes to a question common in the investment world for all groupings of assets. If Ethereum does well in a given month or year, is it because *Ethereum* is special in relation to its coin colleagues or is it because cryptocurrencies *as a whole* did well? In more statistical words, how much of the variation in individual cryptocurrencies could be described by variation in the universe of cryptocurrencies as a whole? And relatedly, is it useful to aggregate cryptocurrencies into an “asset class” or grouping? Before answering this ourselves, we review prior related studies on the matter.

5.1 (a) Existing Literature

Broadie (2012) conducts PCA on the price returns of assets that are generally accepted to rise and fall in a parallel fashion as a result of changes in prevailing and expected future interest rates: on-the-run Treasury securities of varying maturities. He finds that not only does the first principal component account for an overwhelming majority of the variation (95%), but also that the loadings of the first component are near-equal with respect to each individual Treasury security. Of course, these findings are not surprising given that the most significant risk to the price of Treasury securities, i.e., cause for variation in their price returns, is a change in the interest rate curve the effect it has on the discounting of cash flows. Unlike with Treasuries, the causes for variation in the return on equities are arguably more nuanced. Feeney and Hester (1967), in one of the first studies of its kind, conduct PCA on the rates of return of the 30 Dow Jones industrial (DJI) stocks over a 50 quarter period and find that the first component accounts for 41% of the variance (the second component accounts for 9%) and loads positively on each of the 30 stocks. They conclude that the returns of individual stocks are “dominated by the tone of the market.” Borrowing their terminology, we can rephrase the question addressed by this section: is there a significant “tone of the market” for cryptocurrencies?

To answer these questions, we first employ a test similar to the one used by Feeney and Hester (1967) to determine whether it is useful to aggregate stocks into industries. If it were useful, we would expect assets in the same industry to enter different components of the market with similar weights. We conduct proportion tests, using a null hypothesis that the correlations of two cryptocurrencies with a component (i.e., their loadings with respect that component) will be of the same sign with a probability of 0.5, with a two-sided alternative hypothesis that the probability is not equal to 0.5. Rejecting this hypothesis would imply that an investor could reduce the risk in their portfolio by spreading out their exposure to cryptocurrencies among multiple cryptocurrencies, just as one might spread out their exposure to equities, as an asset class, among many stocks. Like Feeney and Hester (1967), we will assume the signs are binomially distributed.

5.2 (b) Proportion test

Using the number of components that it takes to explain more than 70% of the variation (3) for the cryptocurrencies, we find that $z = 2.39$, which is statistically significant at the 5% level.

Therefore, we reject the null hypothesis and conclude that investors *could* reduce their risk in their cryptocurrency portfolio by spreading out their exposure among multiple cryptocurrencies.

As a reference point, we compare the results of the proportions test on the cryptocurrencies to the results we would get by applying the same test on a different set of assets. As a comparison, we chose the top seven holdings of the Energy Select Sector SPDR Fund (Bloomberg Ticker: XLE), which "seeks to provide an effective representation of the energy sector of the S&P 500 Index." Discourse around the energy sector often implies that the stocks are highly correlated with one another, with banks such as Goldman Sachs and JP Morgan often making sector-wide recommendation calls (i.e., buy or sell the "energy stocks") depending on the tides of oil prices. We find that $z = 1.89$, which is not statistically significant at the 5% level. Therefore, we accept the null hypothesis and conclude that investors could not as effectively reduce their risk in their energy holdings by spreading out their exposure among multiple energy stocks. In short, the pundits appear to be onto something in implying that the energy stocks tend to move in tandem with one another to a strong degree.

5.3 (c) Implications

If the price returns of cryptocurrencies were highly correlated, then investing in more than one cryptocurrency would provide no advantage from the perspective of building a diversified portfolio to reduce overall variance (unless, perhaps, you were to short one or more of the other cryptocurrencies). However, the proportions test performed above confirmed that cryptocurrencies are not highly correlated. To address how this information could be used by an investor, we revisit the loadings of the components on the different cryptocurrencies,

	rip	btc	eth	ltc	mon	nem	dash
component							
1	0.844	0.544	0.583	0.742	0.224	0.793	0.284
2	-0.278	-0.176	0.596	-0.363	0.383	-0.074	0.793
3	0.102	-0.296	0.198	0.063	-0.853	0.036	0.266
4	-0.250	0.760	0.006	-0.198	-0.273	-0.039	0.114
5	0.011	0.071	-0.171	0.459	0.042	-0.481	0.293
6	-0.009	-0.043	-0.485	-0.059	0.018	0.321	0.346
7	-0.373	-0.044	0.058	0.250	-0.006	0.171	-0.051

Table 5: Component loadings for the crypto currencies

A risk-averse investor who wanted to invest in cryptocurrencies may want to purchase exposure to dash, nom, and ltc. There is a decent amount of flexibility, but the point of the exercise is to purchase uncorrelated variation, which helps reduce portfolio variance (Feeney and Hester, 1967). What such a risk-averse investor would likely not want to do, for example, is invest in only rip and ltc, whose loadings on the first three components all share the same sign.

Of course, this analysis will be useful in future years only if the covariance matrix remains relatively unchanged over time. It may be the case that cryptocurrencies that are relatively uncorrelated with each other in recent history may become more correlated in the future, eliminating the purported diversification benefit believed to exist today. Although we cannot look into the future, we can investigate the inter-temporal consistency of the covariance matrix by splitting the recent history we have available into two samples.

5.4 (d) Inter-temporal consistency of covariance matrix

The proportion of variance is significantly higher in the second time period. Building a regression model to forecast the loadings in the second time period based on the loadings from the first time period is unsuccessful.

Unfortunately, then, any portfolio considerations formed should be utilized with caution given the changing nature of the covariance matrix.

[to-do next: try month-to-month samples]