Notes on Principle Component Analysis

Nathaniel

I have long wanted to dive into the intricacy of Principle Component Analysis (PCA). This note serves as my attempt to do so. In the following, I will explore what PCA is and apply it on interest rate dynamics as an example using Matlab, Stata, R and SAS.

PCA is one of the most popular techniques in unsupervised learning where only data on independent variables (features) are given. The key idea behind PCA, is to use orthogonal projections to find lower dimensional representations of data that retain as much information as possible. It is also often used as a tool in exploratory data analysis and as a dimension reduction technique.

To motivate the study of PCA, I set out to apply PCA on interest rate dynamics. The dataset, taken from Miller (2013), is daily U.S. government rates from March 2000 through August 2000 with six variables representing maturities of 1, 2, 3, 5, 10, and 30 years. I hope to demonstrate the following observation made by Dowd (2005) P.120

> "The standard financial example is where the original variables might be different spot (or interest) rates across the maturity spectrum, and where the first three principal components are reported to explain over 95% of spot-rate behaviour... the first principal components can be interpreted as reflecting the level of the spot-rate curve, the second can be interpreted as reflecting its steepness; and the third can be interpreted as reflecting its curvature. We can therefore model the spot rates using only a small number of spot-rate principal components—which shows that PCA can be useful for reducing the dimensionality of a multivariate problem."

Figure 1 recreates the example from Miller (2013) Exhibits 9.17 (P.190) and Excel workbook (Chapter_09_PCA_Interest_Rates). It shows that using only the first three of the six principle component, one can approximate the actual on-year yield curve very closely. PCA can thus serves "as a basis for an interest rate model or as the basis for a risk report. A portfolio's correlation with these PCs might also be a meaningful risk metric." (Miller 2013, Page 189).

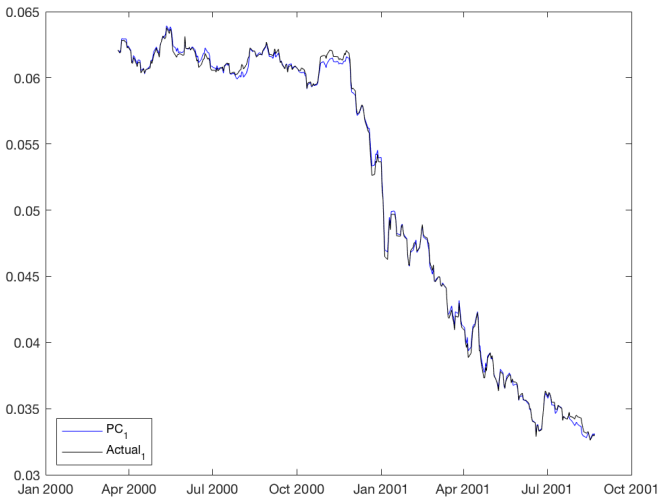

Figure 1: Actual and Approximate 1 Rates (Miller, 2013, Exhibit 9.17)

The Mathematical Idea behind PCA

Before diving into PCA, it would be best to review the following concepts in linear algebra:

- Inner products for computing the lengths, distances, and angles between vectors. An inner product is defined as a symmetric, positive definite, bilinear mapping. An example would be $\beta(x^T Ay)$.
- The big picture (from MIT 1806) of linear algebra—column space, left null space, row space and null space from the equation $Ax = b$.
- Orthogonal projections of data onto lower dimensional subspaces. Recall $P_X = X(X^T X)^{-1} X^T$ and $M_X = I - P_X = I - X(X^T X)^{-1} X^T$.
- Orthonormal basis. The orthogonal square matrix Q and it properties such as $Q^T Q = I$, $Q^T = Q^{-1}$. Gram-Schmidt process. QR decomposition to decompose any real square matrix A into Q and an upper triangular matrix R. The projection matrix becomes QQ^T. The projection solution is found from $Q^T Qx = Q^T b$.
- Eigendecomposition (Spectral Decomposition). Covariance matrix as positive definite symmetric matrix (all eigenvalues are positive and all pivots are positive). $A = UDU^T$ (Rotate Strech Rotate), where D is a diagonal matrix. The entries of the diagonal of D are the eigenvalues of A. The column vectors of U are the eigenvectors of A and they are orthonormal.
- Singular value decomposition (SVD). Spectral Decomposition is one of the special cases of SVD where $U = V = Q$ if A is symmetric positive definite. SVD transform a set of orthogonal bases in one space (row space) to another (column space) while keeping orthogonality of the two sets of bases intact. The SVD theorem states:

  $$A_{nxp} = U_{n\times n} S_{n\times p} V^T_{p\times p}$$

  where $U^T U = I_{n\times n}$ and $V^T V = I_{p\times p}$ (i.e. U and V are orthogonal; the columns of U are the left singular vectors; S has singular values and is diagonal; and V^T has rows that are the right singular vectors. The SVD represents an expansion of the original data in a coordinate system where the covariance matrix is diagonal.

  Calculating the SVD consists of finding the eigenvalues and eigenvectors of AA^T and $A^T A$. The eigenvectors of $A^T A$ make up the columns of V , the eigenvectors of AA^T make up the columns of U. Also, the singular values in S are square roots of eigenvalues from AA^T or $A^T A$. The singular values are the diagonal entries of the S matrix and are arranged in descending order. The singular values are always real numbers. If the matrix A is a real matrix, then U and v are also real.

  PCA is simply an application of the SVD. **The principal components are equal to the right singular values (The principal components of X are the columns of V)** if you first scale the data by subtracting the column mean and dividing each column by its standard deviation (that can be done with the `scale()` function in R). That is, the first column of V is the first principal component and so on and so forth. Using the SVD, the score matrix T is: $T = AX = USV^T V = US$ and one can use the first k column of T (where $k < m$) for dimensionality reduction purpose.

  Also note that the singular value decomposition automatically picked up the differences in the row and column means of a clustered data matrix (see Figure 2). From the R package `swirl`: "Here we again show the clustered data matrix on the left. Next to it we've plotted the first column of the U matrix associated with the scaled data matrix. This is the first LEFT singular vector and it's associated with the ROW means of the clustered data. You can see the clear separation between the top 24 (around -0.2) row means and the bottom 16 (around 0.2). The rightmost display shows the first column of the V matrix associated with the scaled and clustered data matrix. This is the first RIGHT singular vector and it's associated with the COLUMN means of the clustered data. You can see the clear separation between the left 5 column means (between -0.1 and 0.1) and the right 5 column means (all below -0.4). As with the left singular vectors, the other columns of V don't show this pattern as clearly as this first one does".

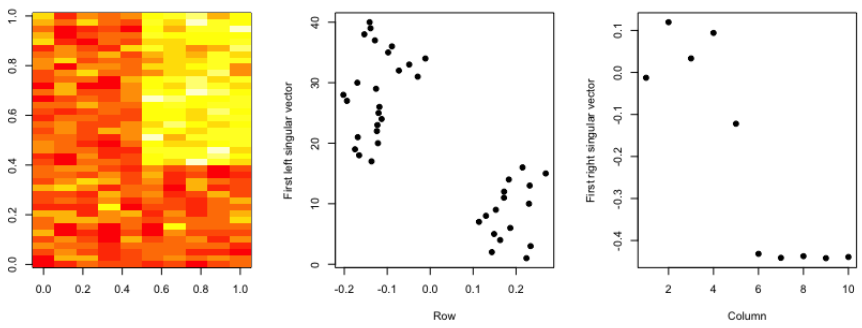

Figure 2

There are at least five different perspectives of PCA that lead to different objectives:

1. minimising the squared reconstruction error,
2. minimising the auto-encoder loss,
3. maximising the mutual information,
4. maximising the variance of the projected data, and
5. maximising the likelihood in a latent variable model

All these different perspectives give us the same solution to the PCA problem. The strengths and weaknesses of individual perspectives become more clear and important when we consider properties of real data. This note focuses on the first and third perspectives.

So what is PCA? In a given vector space, there is potentially an infinite number of orthonormal bases. The question is whether there are any orthonormal bases more 'special' than others? In PCA, a basis is chosen so that the first vector in the basis (principal component), explains as much of the variance in the data, as possible. In higher dimensions, each successive principal component explains the maximum amount of variance in the residual data after taking into account all of the preceding components (orthogonal complement to its 'previous' principle subspace). Just as the first principal component explained as much of the variance in the data as possible, the second principal component explains as much of the variance in the textitresiduals and so on while **keeping the covariance (or correlation) between principle components zero**.

Borrowing from the notion in Tsay (2010) P.483 and Dowd (2005) P.119, the above paragraph on maximising the variance of the projected data can be translated to the following:

$$\begin{aligned} max_w : Var(y_i) &= \boldsymbol{w_i^T \Sigma_r w_i} \qquad i = 1, ..., k, \\ min_w : Cov(y_i, y_j) &= \boldsymbol{w_i^T \Sigma_r w_i} = 0, \qquad i \neq j \end{aligned}$$

subject to the constraints that the length of the orthonormal basis is 1, that is $w_i^T w_i = \sum_{j=1}^{k} w_{ij}^2 = 1$. The problem at hand could be solved by Spectral Decomposition. and the solution is that:

The ith principal component of $\mathbf{r}$ is $y_i = \boldsymbol{e_i' r} = \sum_{j=1}^{k} e_{ij} r_j$ for $i = 1, ..., k$. Moreover,

$$\begin{aligned} Var(y_i) &= \boldsymbol{e_i' \Sigma_r e_i} = \lambda_i, \qquad i = 1, ..., k, \\ Cov(y_i, y_j) &= \boldsymbol{e_i' \Sigma_r e_i} = 0, \qquad i \neq j \end{aligned}$$

To understand how the the above equations are set up, let us dive into the mathematics of PCA (This is related to data compression). From the perspective of maximizing variance, see ISLR (equation 10.3) page

376. Here we will adopt the perspective of minimizing reconstruction error. Given data centred with mean zero and represented by matrix X_n with D column of $x_i...x_d$, X_n can be represented by orthonormal basis matrix B which consists of orthonormal bases $b_i...b_D$.

$$X_n = \sum_{i=1}^{D} \beta_{in} b_i$$

$$X_n = \sum_{i=1}^{M} \beta_{in} b_i + \sum_{i=M+1}^{D} \beta_{in} b_i$$

The key idea in PCA is to find a lower dimensional representation of X_n, denote as $\tilde{X_n}$ that can be expressed using fewer basis vectors. $\tilde{X_n}$ is still a D-dimensional vector, it lives in an M-dimensional subspace of $\mathbb{R}^{\mathbb{D}}$ and only M coordinates: $\beta_{n1}...\beta_{n_M}$ are necessary to represent it, which leads to the deletion of the second component

$$\tilde{X}_n = \sum_{i=1}^{M} \beta_{in} b_i$$

We want to find parameters β_{in} and orthonormal basis vectors $b_i...b_D$ which span the principle subspace, such at the loss function—the average squared reconstruction error is minimised. β_{in} can be interpreted as *weights (loadings)* of the orthonormal bases. But if one was allowed to make those weights (loadings) as big as he/she wanted, one could make the variance of as big as possible. The natural constraint is to normalize β_{in} so the sum of their squares is 1. The average squared reconstruction error is denoted as J optimized through standard derivative procedure (including multivariate chain rule). The whole procedure is detailed in the online course, Mathematics for Machine Learning: PCA, offered by the Imperial College London (week 4). The key equations to note are:

$$min_{\tilde{x}_n} : J = \frac{1}{N} \sum_{n=1}^{N} \|x_n - \tilde{x}_n\|^2$$

$$\frac{\partial J}{\partial \beta_i n} = \frac{\partial J}{\partial \tilde{x}_n} * \frac{\partial \tilde{x}_n}{\partial \beta_i n}$$

$$\frac{\partial J}{\partial \beta_i n} = -\frac{2}{N}(x_n - \tilde{x}_n)^T * b_i = 0$$

$$\Rightarrow \beta^*_{in} = x_n^T b_i$$

Substituting $\beta^*_{in} = x_n^T b_i$ into $x_n - \tilde{x}_n$ and reformulating the J gives

$$J = \sum_{j=M+1}^{D} b_J^T S b_j = trace\left(\sum_{j=M+1}^{D} b_i b_j^T \right) S$$

where S is the data covariance matrix.

In a 2 dimensional example, the Lagrangian would be set up as follow:

$$L = b_2^T S b_2 + \lambda(1 - b_2^T b_2)$$

Setting the partial derivatives equal to zero and solving for solutions yield:

$$J = \lambda$$

Using Lagrange multipliers generally, it could be shown that

$$J = \sum_{j=M+1}^{D} \lambda_j$$

and

$$\tilde{X} = B \underbrace{B^T x}_{coordinate}$$

In other words, J, the average reconstruction error is minimised if we choose the basis vectors that span the ignored subspace to be the eigenvectors of the data covariance matrix that belong to the smallest eigenvalues. This equivalently means that the principal subspace is spanned by the eigenvectors belonging to the M largest eigenvalues of the data covariance matrix. If the basis vectors are picked in this way, the reconstruction error is minimized and equal to the sum of $D - M - 1$ smallest eigenvalues of the (full) data covariance matrix.

This nicely aligns with properties of the covariance matrix. The eigenvectors of the covariance matrix are orthogonal to each other because of symmetry and the eigenvector belonging to the largest eigenvalue points in the direction of the data with the largest variance and the variance in that direction is given by the corresponding eigenvalue. Similarly, the eigenvector belonging to the second largest eigenvalue points in the direction of the second largest variance of the data and so on.

One of the problem in using PCA is that PCA is prone to missing data which is not unusual. One way to work around this problem is called imputing the data. "This uses the k nearest neighbors to calculate a values to use in place of the missing data. One may want to specify an integer k which indicates how many neighbors you want to average to create this replacement value" (from `swirl` package in R).

In summary, PCA involves the following steps:

1. Standardize the raw data (correlation is still intact).
2. Calculate a covariance matrix of the standardized data.
3. Decompose the covariance matrix to obtain eigenvalues and eigenvectors.
4. Project data onto the principal subspace spanned by the first n eigenvectors ranked according to their eigenvalues (largest to smallest).

A simple 2-vector example in Matlab shows that PC linearly transform data:

Figure 3 (right) shows that the largest PC (x) spans the direction that maximizes the variance of the projected data, and the smallest PC (y) maximizes that variance of the residual. In effect, PCA linearly transforms the data points we had. Quoting from ISLR P.379

> The appeal of this interpretation is clear: we seek a single dimension of the data that lies as close as possible to all of the data points, since such a line will likely provide a good summary of the data ... The notion of principal components as the dimensions that are closest to the n observations extends beyond just the first principal component. For instance, the first two principal components of a data set span the plane that is closest to the n observations, in terms of average squared Euclidean distance.

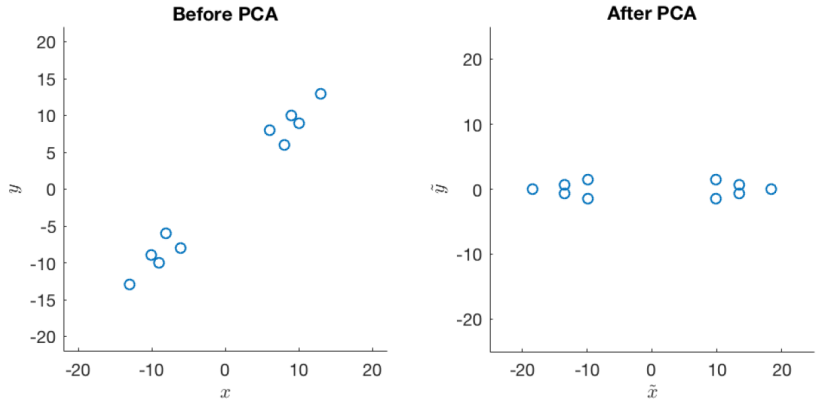

Figure 3

An Example using PCA on Interest Rate Dynamic

Principal components and singular values may mix real patterns. So finding and separating out the real patterns require some detective work. Let us return to the interest rate dynamic example from Miller (2013). From Matlab (see `PCA_Practice.m`), we can derive the eigenvalues as below

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
5.39	0.56	0.041	0.004	0.002	0.0006

the scoring coefficients (or equivalently loading, orthonormal eigenvectors) matrix is:

0.3910	−0.5335	0.6101	0.3367	0.2260	−0.1602
0.4220	−0.2630	−0.03014	−0.3087	−0.2675	0.76476
0.4268	−0.1631	−0.1981	−0.3562	−0.4949	−0.6164
0.42853	0.01134	−0.4604	−0.1798	0.7538	−0.0595
0.4186	0.2949	−0.3152	0.7555	−0.2486	0.0760
0.3576	0.7296	0.5255	−0.2473	0.0469	−0.00916

with the constraint that the norms of eignvector is equal to 1 (for instance, the sum of squares of the first eigenvector is: $0.391^2 + 0.4221^2 + 0.4269^2 + 0.4286^2 + 0.4187^2 + 0.3574^2 \approx 1.00003543$)

And the proportions of the total variance explained by each principle component are

$\lambda_1/\sum\lambda$	$\lambda_2/\sum\lambda$	$\lambda_3/\sum\lambda$	$\lambda_4/\sum\lambda$	$\lambda_5/\sum\lambda$	$\lambda_6/\sum\lambda$
0.89837	0.99198	0.99884	0.99953	0.99989	1.0000

The scree plot in Figure 4 presents individual and cumulative proportion of the total variance explained by the first two principle components. Along with the third principle component, 0.9988% of total variance is explained . With that, we can reasonably argue that they are representative summary of the yield curve data. But the most important question is: how do we interpret them?

Figure 5 plots the three principle components. The takeaway is that they demonstrates the prototypical patterns-shifting, tilting, and twisting of the yield curve! Or paraphrasing from Dowd (2005)'s words, the first principle represents the level of the spot-rate curve, the second the steepness (I additionally think of it as duration in derivative pricing) and third the curvature (convexity). As note by Miller (2013) P189-190:

> [The first principle component] This flat, equal weighting represents the shift of the yield curve.

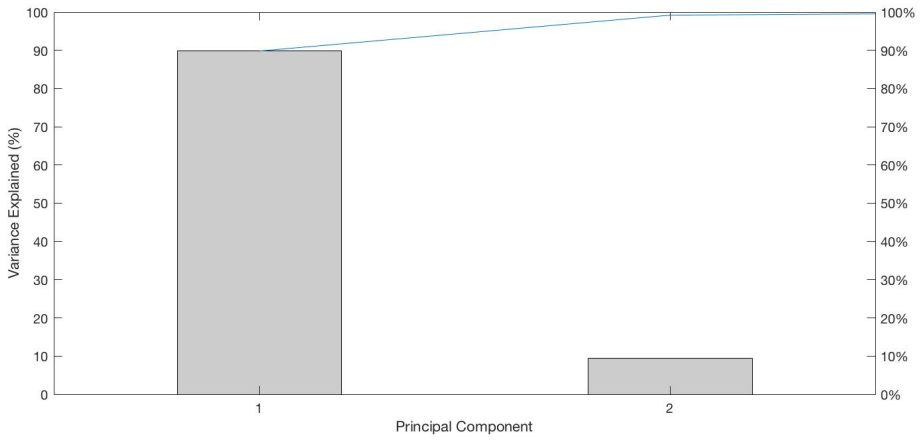

Figure 4: The first two principle components explains over 90% of the variance.

A movement in this component increases or decreases all of the points on the yield curve by the same amount (actually, because we standardized all of the data, it shifts them in proportion to their standard deviation). Similarly, the second principle component shows an upward trend. A movement in this component tends to tilt the yield curve. Finally, if we plot the third principle component, it is bowed, high in the center and low on the ends. A shift in this component tends to twist the yield curve ...

We could multiply all of the elements in one column of the PC matrix ... The justification for doing this is purely aesthetic ...

Because the first three principle components explain so much of the dynamics of the yield curve, they could serve as a basis for an interest rate model or as the basis for a risk report. A portfolio's correlation with these principle components might also be a meaningful risk metric.

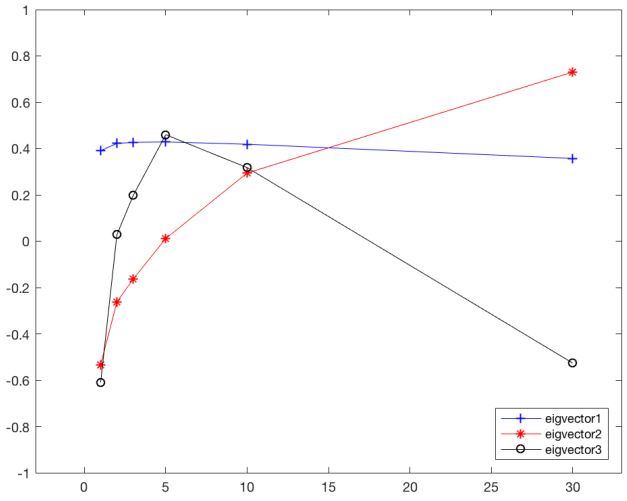

Figure 5: First Three Principal Components' loadings of the Yield Curve (Miller, 2013, Exhibit 9.16)

Figure 6 plots the first two principal component score vectors which give the coordinates ($B^T x$) of the projection of the observations onto the plane, with the variance in the plane being maximized. The idea is akin to Figure 10.1 and 10.2 in ISLR (page 377, 380). One can further plot the first two principal component loading vectors (see Figure 8) onto Figure 6 to create a biplot which will better display both the principal component scores and the principal component loadings for weights' interpretation.

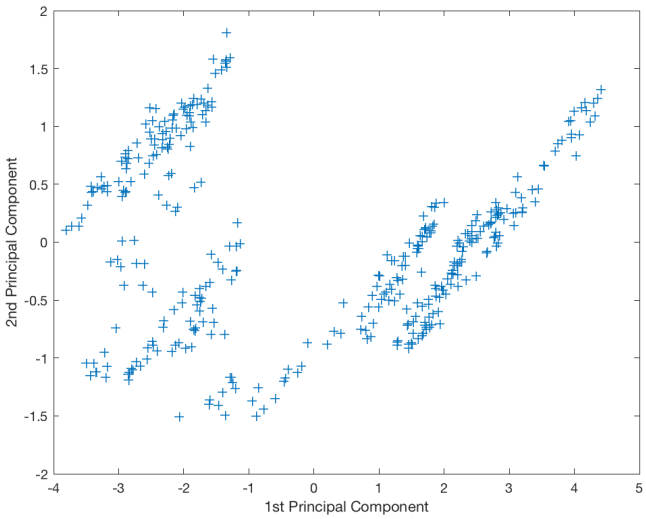

Figure 6: Relationship of the first two principle component scores

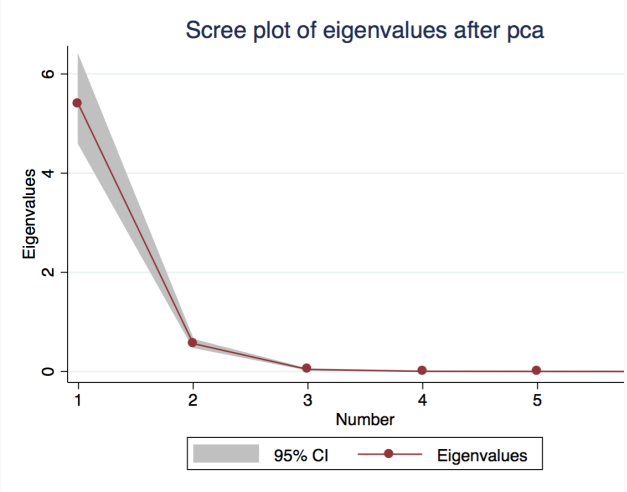

Figure 7

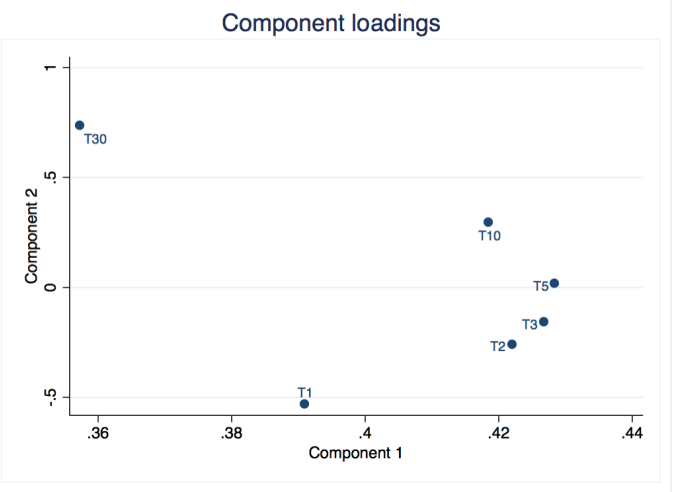

Figure 8

The loading plot (Figure 8) created by Stata examines the first two principle components loadings for the daily U.S. government rates dataset. To see the relationship between loading and eignevector, see url. From the first component, we see that U.S. government rate with 30-year maturity is separated from others. Next comes the 1-year maturity which is in the "middle", potentially highlighting the differences of the two variables from others. From the second component, the difference between the government rate with 1-year and the 30 year of maturity again stands out. That the second principle component could be interpreted as steepness of the yield curve can be seen from the ascending order of the variables from -0.5 to 0.8 (component 2). Also note that ordering of the second principle component is very similar to the second eignvector in Figure 5. This stems from the definition of loadings.

Reference

Dowd Kevin (2005) Measuring Market Risk, Wiley, 2nd Edition

Ruey S. Tsay (2010) Analysis of Financial Time Series, Wiley, 3th Edition

G.James, D.Witten, T.Hastie and R.Tibshirani (2013), An Introduction to Statistical Learning, with applications in R (ISLR), Springer

Michael B. Miller (2013) Mathematics and Statistics for Financial Risk Management, Wiley, 2nd Edition, ISBN: 978-1-118-75029-2

Hull, John. (1989). Options, futures, and other derivative securities. Englewood Cliffs, N.J.: Prentice Hall.