Principal Components of a Multivariate Distribution

2019

X. HUO

Agenda

- Motivation & preliminary
- Mathematical background
- Dimension reduction
- Applications and an example

- Reading: Chapter 18 in Ruppert

Motivation

Correlated financial systems

- Many financial markets are characterized by a high degree of collinearity between returns.
- We often need to generate a large set of scenarios based on movements in many risk factors for risk management or pricing models. The computation is expensive due to the high dimensions of factors.
- Variables are highly collinear when there are only a few important sources of information in the data that are common to many variables.
  - Term structures of interest rates
  - Implied volatility of different same underlying assets
  - Futures of different maturities on the same underlying assets……

Idea: Extract the most important uncorrelated sources of variation in a multivariate system --- principal component analysis (PCA).

Motivation

- Reducing dimensionality
  - Introduce a data transformation tool for correlated financial systems.
  - The reduction in dimensionality is achieved by taking only the first few principal components.
  - This transformation significantly reduces the computation time.
- The orthogonal property
  - The sample covariance matrix is not always positive definite (the determinant is about 0). This can be caused by high linear dependency of one variable to another, or large amounts of missing data. In this case, the sample covariance matrix can not be used for portfolio optimization as the singular matrix does not have an inverse.
  - Principal components are orthogonal, and their covariance matrix is diagonal. The principal components can be transformed into a positive definite covariance matrix.

Preliminary

The data input to PCA must be stationary.

- Prices, rates or yields are generally non-stationary and so they will have to be transformed into returns, before PCA is applied.

The returns need to be normalized before the analysis.

- Otherwise, the first PC will be dominated by the input variable with the greatest volatility.

PCA is based on an eigenvalue and eigenvector analysis of the correlation (or covariance) matrix.

Mathematical Background

Let $\mathbf{x}=(x_1,\ldots,x_p)^T$ be a random vector with mean μ and covariance matrix V.

First principal component:

$$\mathbf{a}_1 = \arg\max \mathsf{Var}(\mathbf{a}'\mathbf{x}) = \arg\max[\mathbf{a}'V\mathbf{a}] \text{ subject to } ||\mathbf{a}|| = 1$$

Introduce the Lagrange multiplier λ:

$$\frac{\partial}{\partial a_i}\{\mathbf{a}^T V\mathbf{a} + \lambda(1 - \mathbf{a}^T\mathbf{a})\} = 0 \text{ for } i = 1, \ldots, p$$

$$\Longrightarrow V\mathbf{a} = \lambda\mathbf{a}$$

This implies that

- λ is an eigenvalue of V
- **a** is the corresponding eigenvector

Mathematical Background

- Second principal component:

$$\mathbf{a}_2 = \arg\max \mathsf{Var}(\mathbf{a}'\mathbf{x}) = \arg\max[\mathbf{a}'V\mathbf{a}]$$

subject to $\mathbf{a}_1'\mathbf{a}_2 = 0$ and $||\mathbf{a}_2|| = 1$.

Introduce the Lagrange multiplier λ_1 and λ_2:

$$\frac{\partial}{\partial a_i}\{\mathbf{a}^T V\mathbf{a} + \lambda_1(1 - \mathbf{a}^T\mathbf{a}) + \lambda_2\mathbf{a}_1'\mathbf{a}_2\} = 0 \text{ for } i = 1, ..., p$$

$$\Longrightarrow V\mathbf{a}_2 = \lambda\mathbf{a}_2, \quad \mathbf{a}_1'\mathbf{a}_2 = 0$$

- Third principal component, …

Mathematical Background

Since V is symmetric, its eigenvalues (solutions of the polynomial equation det(V-λ I)=0) are real and can be ordered as λ_1 ¸ … ¸ λ_p.

They are all nonnegative since V is nonnegative definite.

Moreover,

$$tr(V) = \lambda_1 + ... + \lambda_p, \quad det(V) = \lambda_1 ... \lambda_p.$$

Let a_j be the eigenvector corresponding to λ_j, then the eigenvectors are orthogonal to each other.

- Definition:
  - a_I^T x is called the i-th principal component of x.

Mathematical Background

Basic Facts:

(a) $V = \lambda_1 \mathbf{a_1 a_1^T} + \ldots + \lambda_p \mathbf{a_p a_p^T}, \quad I = \mathbf{a_1 a_1^T} + \ldots + \mathbf{a_p a_p^T}$

(b) $\sum_{i=1}^{p} Var(x_i) = tr(V) = \lambda_1 + \ldots + \lambda_p.$

(c) $Var(\mathbf{a_i^T x}) = \lambda_i$

(d) We hope that only a few principal components account for most of the overall variance.

$(\sum_{i=1}^{k} \lambda_i)/tr(V)$ is near 1 for some small k.

(e) Factor loadings are columns giving the elements of the column vectors a_i for the principal components $a_i^T x$.

Mathematical Background

$(a)\ V = \lambda_1 \mathbf{a_1 a_1^T} + \ldots + \lambda_p \mathbf{a_p a_p^T}, \quad I = \mathbf{a_1 a_1^T} + \ldots + \mathbf{a_p a_p^T}$

Proof:

$$V \cdot (\mathbf{a}_1 \ldots \mathbf{a}_p) = (\mathbf{a}_1 \ldots \mathbf{a}_p) \begin{pmatrix} \lambda_1 & & \\ & \ldots & \\ & & \lambda_p \end{pmatrix}$$

$$\Longrightarrow \quad V = (\mathbf{a}_1 \ldots \mathbf{a}_p) \begin{pmatrix} \lambda_1 & & \\ & \ldots & \\ & & \lambda_p \end{pmatrix} \begin{pmatrix} \mathbf{a}_1^T \\ \ldots \\ \mathbf{a}_p^T \end{pmatrix}$$

$$\Longrightarrow \quad V = (\mathbf{a}_1 \lambda_1 \ldots \mathbf{a}_p \lambda_p) \begin{pmatrix} \mathbf{a}_1^T \\ \ldots \\ \mathbf{a}_p^T \end{pmatrix} = \sum_{i=1}^{p} \mathbf{a}_i \lambda_i \mathbf{a}_i^T$$

Mathematical Background

Suppose $x_1,\ldots,x_n$ are a sample of n independent observations from a multivariate population with mean μ and covariance matrix V.

$$\hat{\mu} = \bar{x} = \sum_{i=1}^{n} x_i/n, \qquad \hat{V} = X^T X/(n-1)$$

$$X = \begin{pmatrix} x_{11} - \bar{x}_1 & \ldots & x_{1p} - \bar{x}_p \\ \ldots & \ldots & \ldots \\ x_{n1} - \bar{x}_1 & \ldots & x_{np} - \bar{x}_p \end{pmatrix} = (\mathbf{X}_1, \ldots, \mathbf{X}_p)$$

- The jth principal component of $\mathbf{X}_1,\ldots,\mathbf{X}_p$ is the linear combination

$$\mathbf{Y}_j = \hat{a}_{1j}\mathbf{X}_1 + \ldots + \hat{a}_{pj}\mathbf{X}_p$$

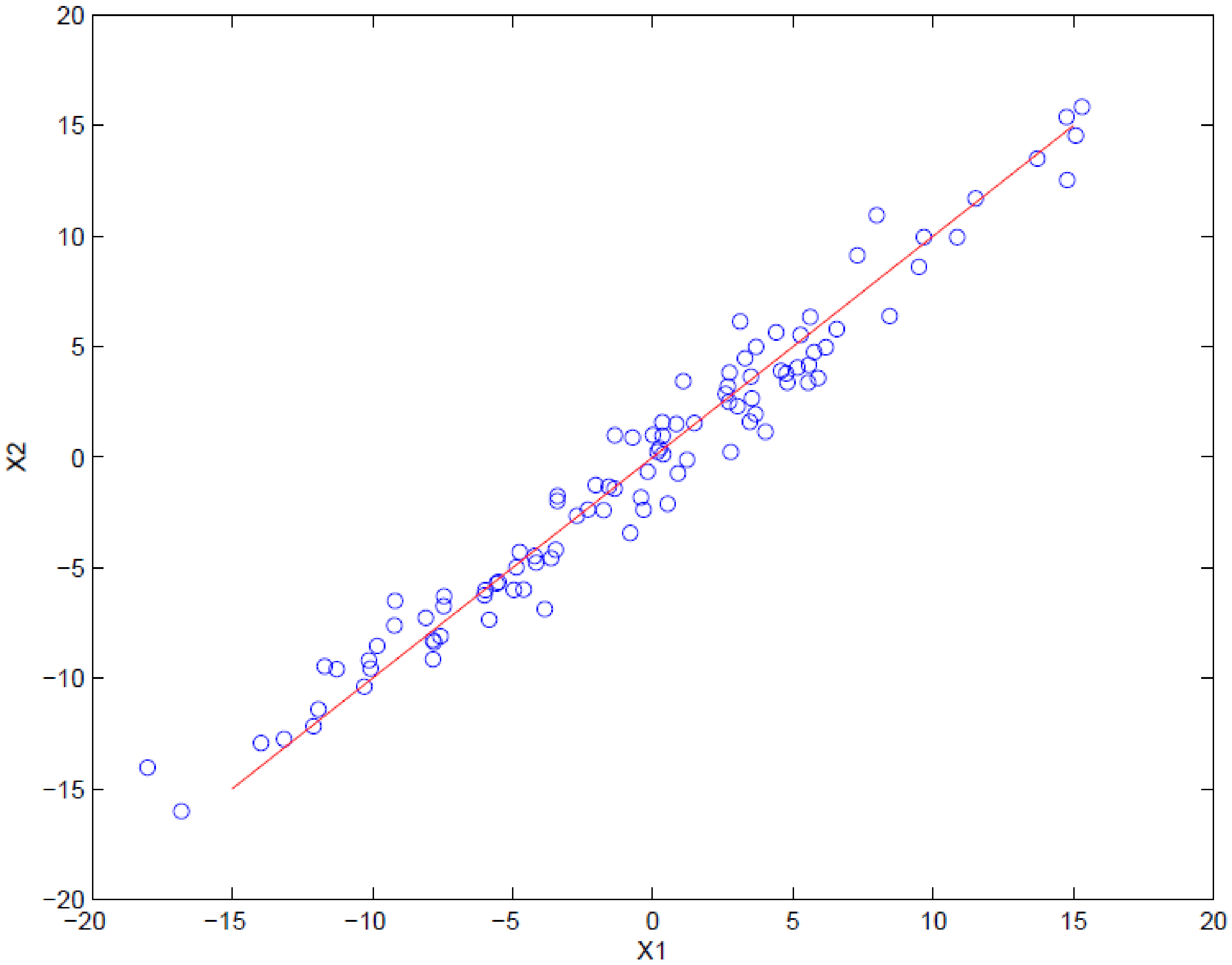
where $\hat{\mathbf{a}}_j = (\hat{a}_{1j}, \ldots, \hat{a}_{pj})^T$ is the eigenvector corresponding to the jth largest eigenvalue $\hat{\lambda}_j$ of the sample covariance matrix $\hat{V}$.

A MATLAB Illustration

Codes

```
n = 100;
y1 = randn(n,2) * [10 0; 0 1];
theta = pi/4;
rotmat = [cos(theta) sin(theta); -sin(theta) cos(theta)];
y2 = y1 * rotmat; clf;
plot(y2(:,1), y2(:,2), 'o');
hold on;
plot([-15 15], [-15 15], 'r');
xlabel('X1')
ylabel('X2')
print -depsc2 pcai01
```

A MATLAB Illustration



Applications

Dimension reduction:

- When p is large, we need to estimate p(p+1)/2 parameters for p by p covariance matrix $\hat{V}$.
- If many λ_i are small, only a few, say k, principal components are involved in the standard error of the estimate $\hat{a}_j$ for $1 \leq j \leq k$.
- Thus, we reduce a high-dimensional estimation problem (with p(p+1)/2 parameters) to a more tractable problem with much fewer parameters.

Term structure of **interest rates**, **implied volatilities**, …

Applications

Multifactor models

- Choose the first k principal components of asset returns as k factors in multifactor models.

  The first principal component is the (normalized) linear combination of asset returns that gives the largest proportion of the variance.
- An alternative to principal components in the choice of factors is factor analysis, which is beyond the scope of this lecture.
  - Factor analysis assumes that there are indeed k factors so that

    $$r_i = \alpha_i + \beta_i^T f + \varepsilon_i \qquad (i = 1, \ldots, N),$$

    in which $f = (f_1, \ldots, f_k)^T$, with the factors f_i yet to be determined, and the ε_i are assumed to be normal.
  - Unlike principal component analysis, factor analysis requires pre-specification of the number of factors and distribution assumptions on ε_i.

Example(1): A Portfolio of Stocks

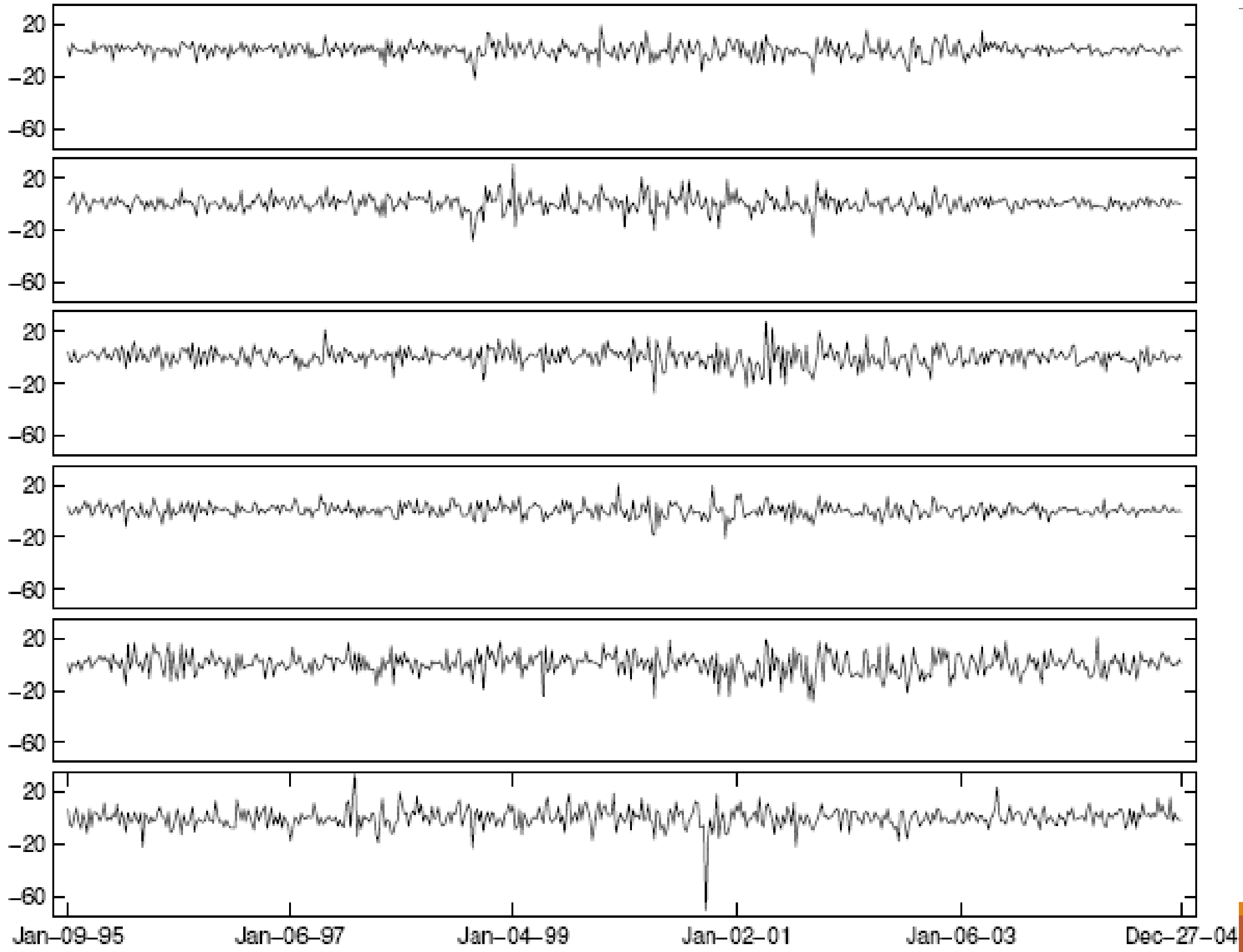
Weekly log returns:

JP Morgan Chase,

Lehman brothers,
Cisco system Inc.,

Microsoft Corp., Sun Microsystems Inc., and

Apple Computer Inc..



Example(1): A Portfolio of Stocks

$$\mathbf{r}' = (\text{JPM, LEH, CSCO, MSFT, SUNW, AAPL}).$$

The sample mean and sample covariance matrix

$$\boldsymbol{\mu} = \begin{pmatrix} 0.2327 & 0.4786 & 0.4458 & 0.3996 & 0.3116 & 0.2140 \end{pmatrix},$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 26.6124 & 17.9980 & 13.5795 & 8.1036 & 14.1662 & 6.2569 \\ 17.9980 & 34.8167 & 18.0034 & 10.5591 & 20.1398 & 9.5653 \\ 13.5795 & 18.0034 & 43.5496 & 16.6841 & 34.5675 & 16.0576 \\ 8.1036 & 10.5591 & 16.6841 & 23.7764 & 18.6738 & 10.8830 \\ 14.1662 & 20.1398 & 34.5675 & 18.6738 & 63.4471 & 19.2954 \\ 6.2569 & 9.5653 & 16.0576 & 10.8830 & 19.2954 & 56.9298 \end{pmatrix},$$

Example(1): A Portfolio of Stocks

The sample correlation matrix

$$\rho = \begin{pmatrix} 1.0000 & 0.5913 & 0.3989 & 0.3222 & 0.3448 & 0.1607 \\ 0.5913 & 1.0000 & 0.4623 & 0.3670 & 0.4285 & 0.2149 \\ 0.3989 & 0.4623 & 1.0000 & 0.5185 & 0.6576 & 0.3225 \\ 0.3222 & 0.3670 & 0.5185 & 1.0000 & 0.4808 & 0.2958 \\ 0.3448 & 0.4285 & 0.6576 & 0.4808 & 1.0000 & 0.3211 \\ 0.1607 & 0.2149 & 0.3225 & 0.2958 & 0.3211 & 1.0000 \end{pmatrix}$$

Example(1): A Portfolio of Stocks

(a) Sample covariance matrix

Eigenvalue	128.8825	45.3876	30.8143	17.7629	14.1168	12.1679
Proportion	51.7326	18.2183	12.3687	7.1299	5.6664	4.8841
Eigenvector	0.2530	-0.2090	0.5420	-0.0909	-0.0044	0.7683
	0.3401	-0.2347	0.6151	-0.2163	0.0269	-0.6352
	0.4828	-0.1571	-0.1414	0.6570	0.5387	-0.0211
	0.2771	-0.0366	-0.0101	0.4697	-0.8362	-0.0432
	0.6054	-0.2099	-0.5385	-0.5368	-0.0882	0.0594
	0.3792	0.9117	0.1339	-0.0680	0.0449	-0.0208

Example(1): A Portfolio of Stocks

78.50% of total variation

General movement of the market

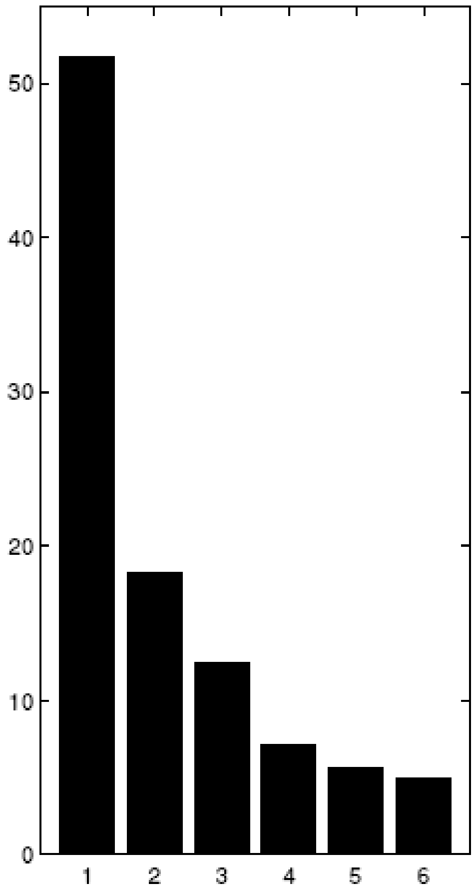
Difference of industrial sectors

(b) Sample correlation matrix

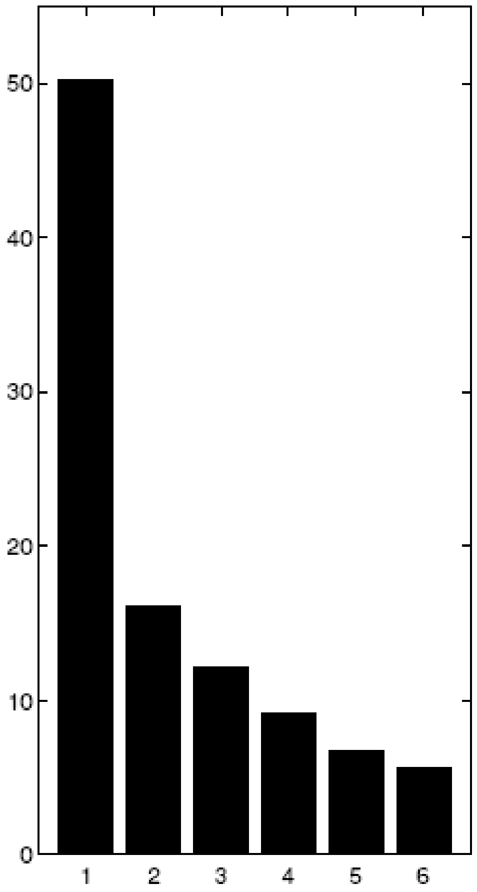
Eigenvalue	3.0130	0.9664	0.7303	0.5524	0.4009	0.3370
Proportion	50.2168	16.1060	12.1722	9.2060	6.6818	5.6171
Eigenvector	0.3831	-0.5729	0.2983	0.1051	0.6429	-0.1085
	0.4220	-0.4524	0.2330	-0.0475	-0.7475	0.0435
	0.4715	0.1357	-0.2985	-0.3034	0.1537	0.7446
	0.4094	0.2106	-0.3237	0.8207	-0.0541	-0.0819
	0.4525	0.1958	-0.3317	-0.4702	0.0255	-0.6520
	0.2835	0.6051	0.7435	-0.0015	0.0259	0.0023

Example(1): A Portfolio of Stocks

Fraction of total variation of each principal component



Covariance matrix



Correlation matrix

Example(2) Swap Rates

Daily swap rates of eight maturities from 7/3/2000 to 7/15/2005. (Relates to Times Series and Interest Rate Markets.)

File name: *d swap.txt*. Downloadable from the web site of the book by Lai and Xing.

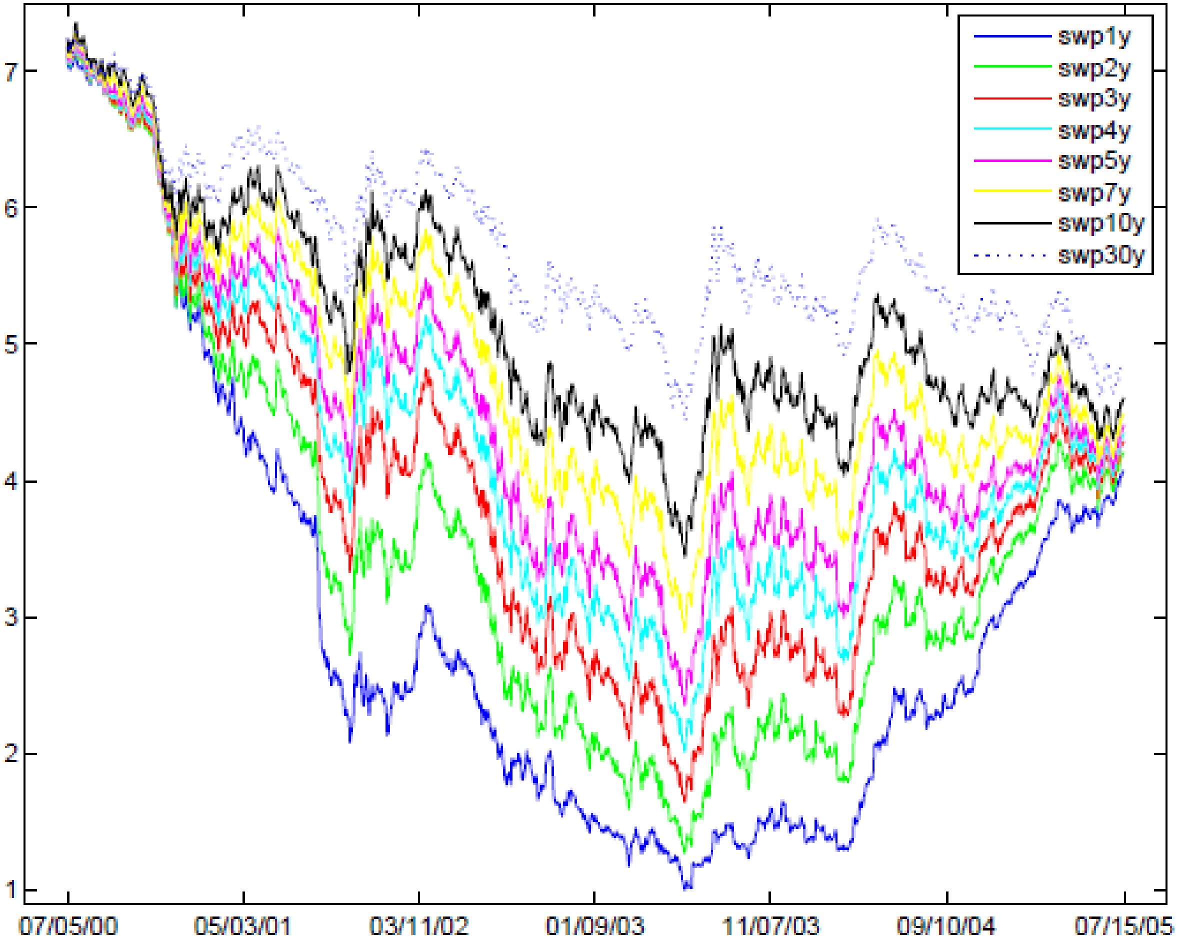
Example(2) Swap Rates

Date file

```
%%% (Data source: Economic, LLC)
%swp1y swp2y swp3y sw4y sw5y sw7y sw10y sw30y
7.1 7.16 7.17 7.17 7.17 7.2 7.24 7.24
7.03 7.06 7.07 7.07 7.08 7.11 7.14 7.16
7.07 7.13 7.14 7.15 7.16 7.19 7.21 7.21
7.01 7.04 7.06 7.06 7.07 7.1 7.14 7.14
7.04 7.09 7.11 7.13 7.14 7.17 7.2 7.19
… …
```

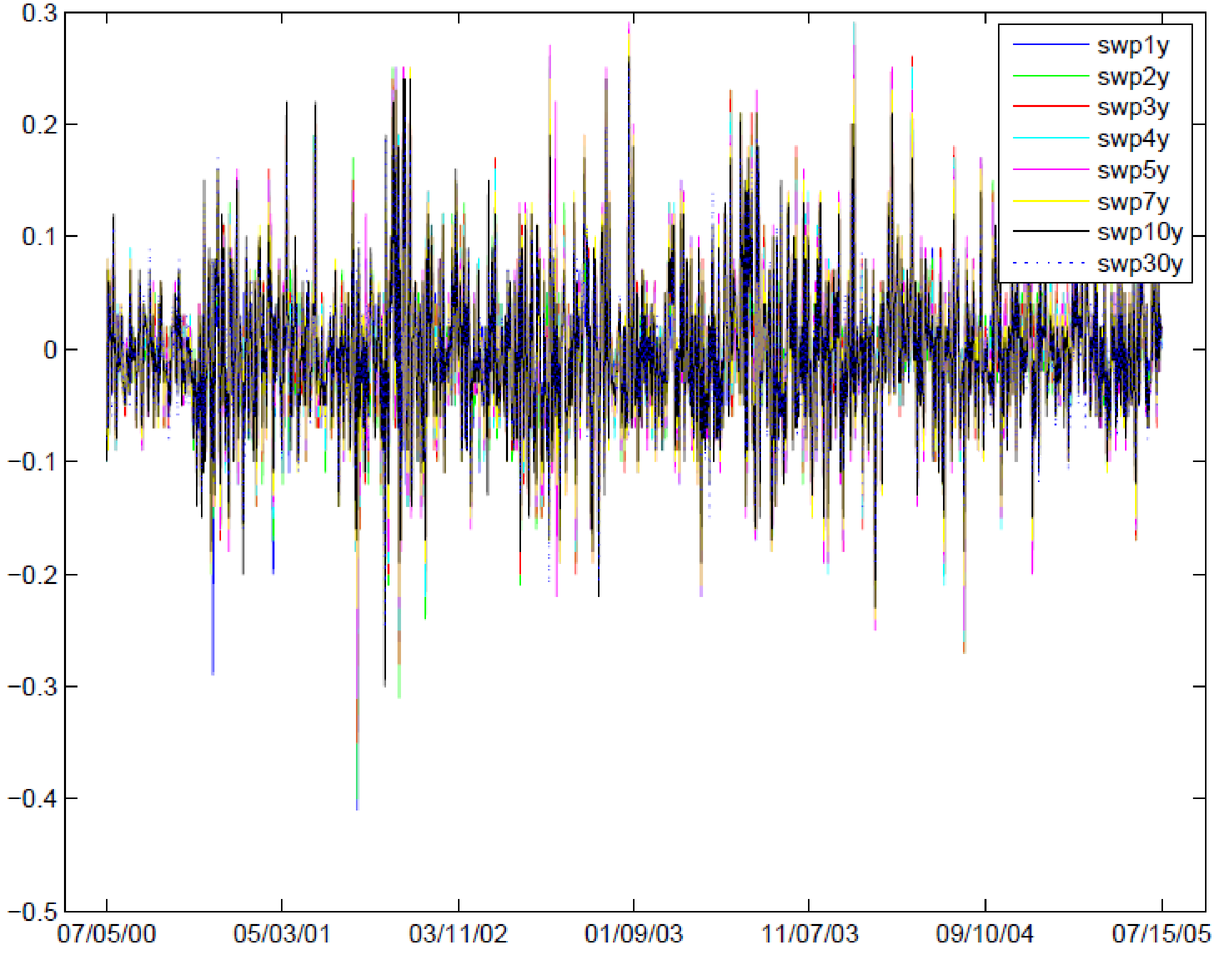
Example(2) Swap Rates

Daily swap rates



Example(2) Swap Rates

Differences of the Daily Swap Rates



Example(2) Swap Rates

Decomposition of Variance

First Three Principal Components

Data with Reduced Dimension

Eigenvectors (Factor Loadings)

Example(2) Swap Rates

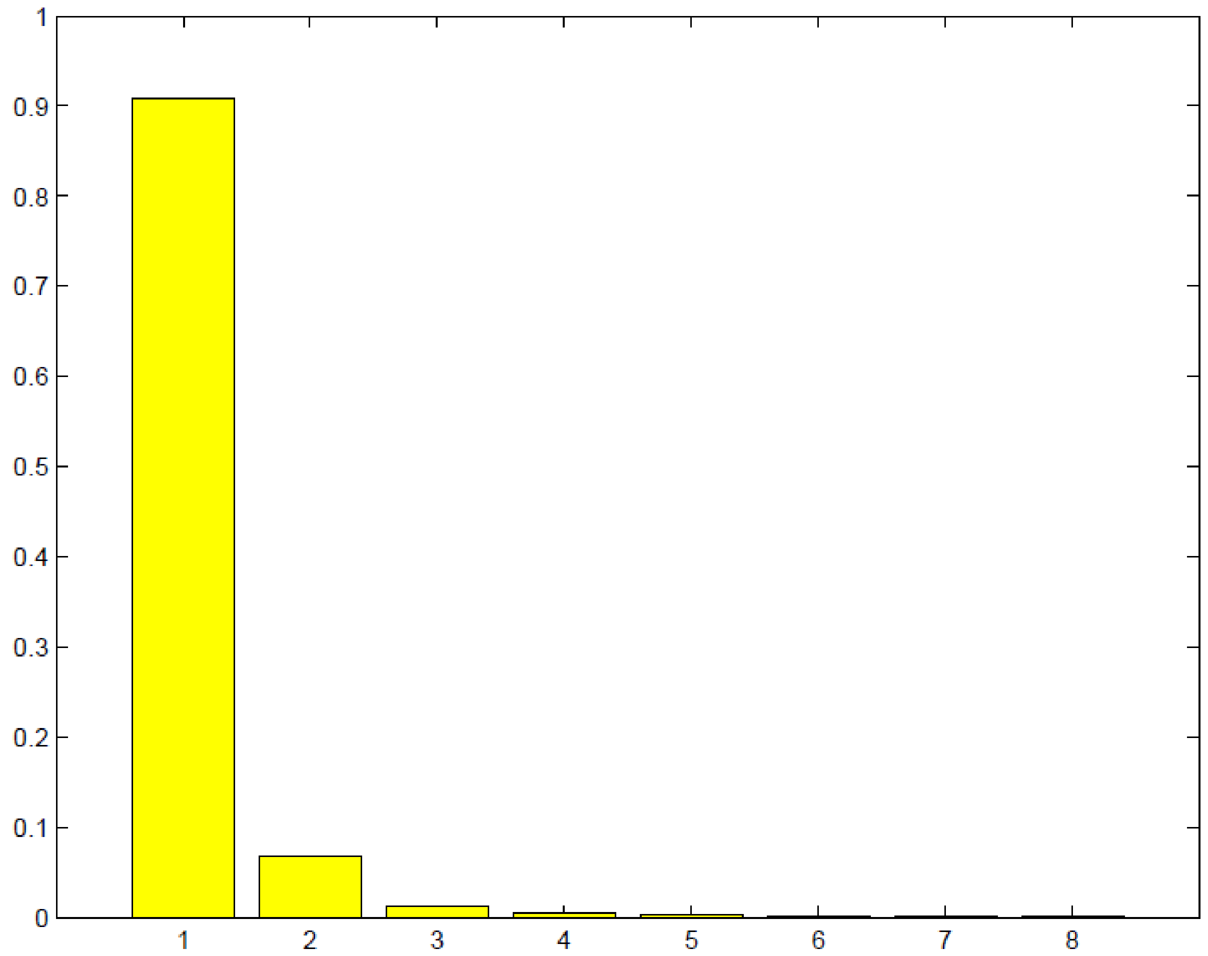
MATLAB codes

```
>> data = load('d_swap.txt');
>> [coeff, eigenvalue, explained] = pcacov(corrcoef(diff(data)));
>> eigenvalue', explained'
ans =
7.2649 0.5477 0.1032 0.0408 0.0221 0.0105 0.0058
ans =
90.8111 6.8459 1.2895 0.5099 0.2757 0.1314 0.0725
```

First three PCs carries most of the variance.

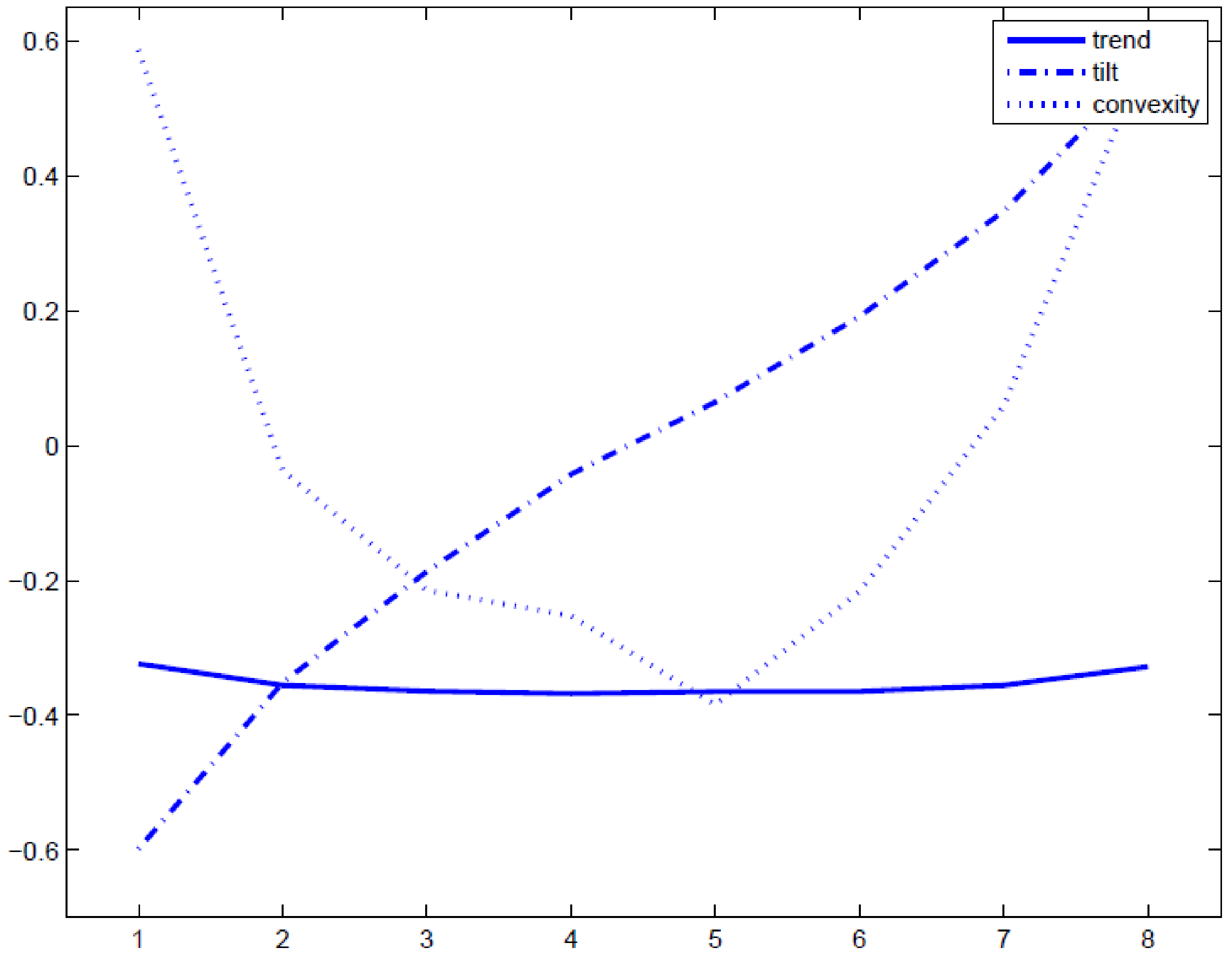
Example(2) Swap Rates

Decomposition of Variance



Example(2) Swap Rates

First Three Principal Components



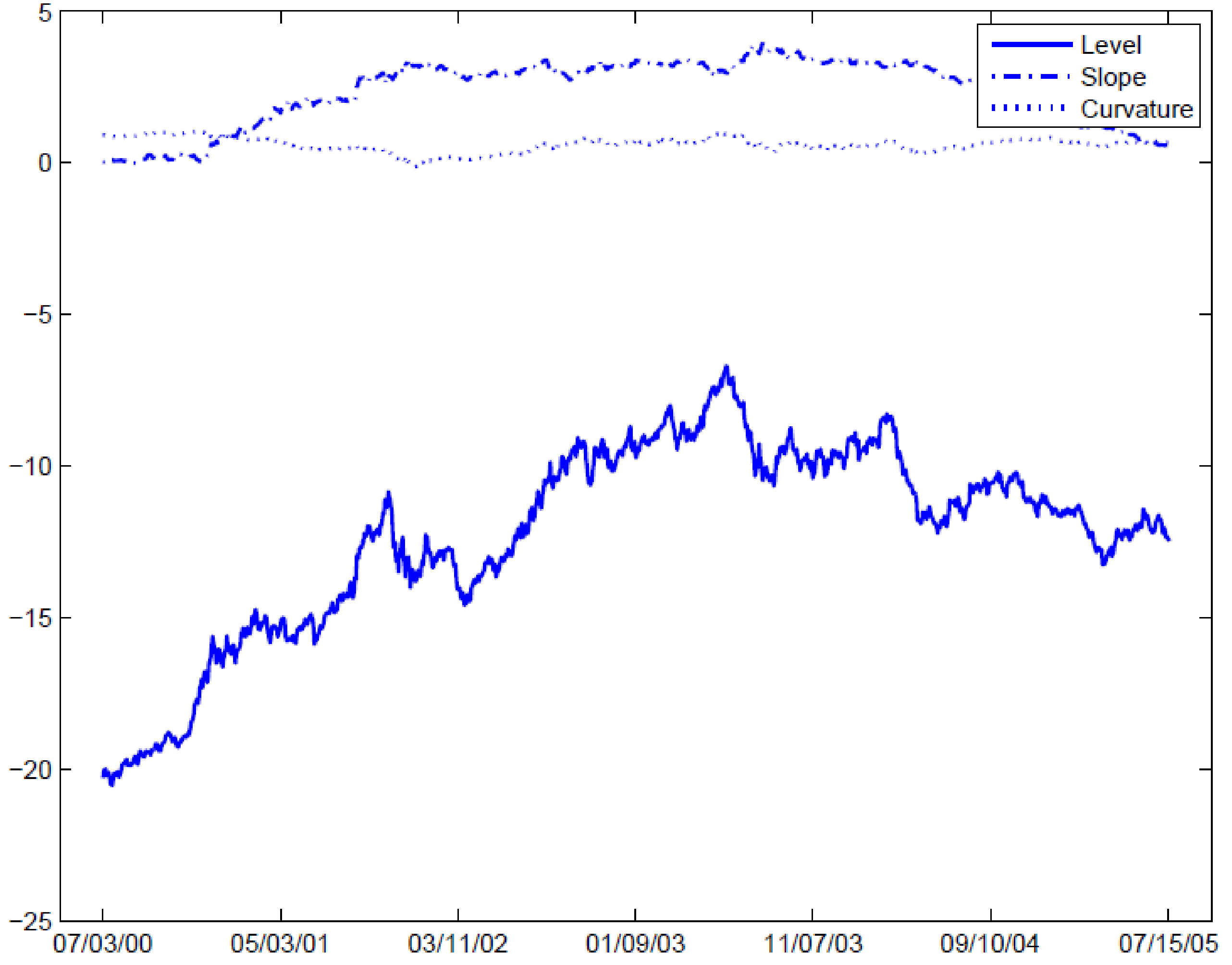
Example(2) Swap Rates

Discussion.

- The first PC is (roughly) the average -- Trend.
- The second PC is the tiltness.
- The third PC measures the curvature.

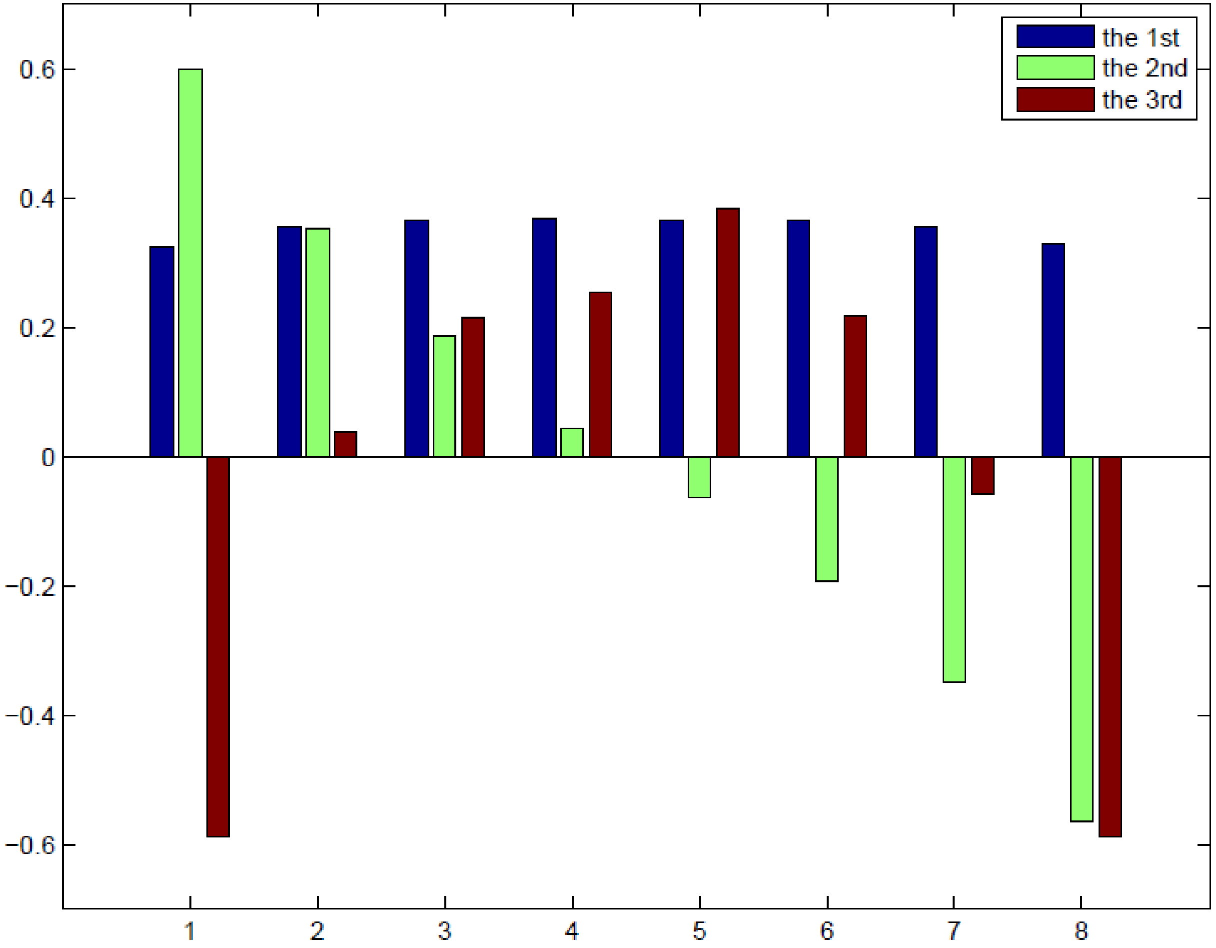
Example(2) Swap Rates

Data with Reduced Dimension



Example(2) Swap Rates

Eigenvectors (Factor Loadings)



Example(2) Swap Rates

MATLAB Source Codes

```
%%% The data file 'd_swap.txt' contains the following daily swap
%%% rates from 7/3/2000 to 7/15/2005.
%%% swp1y swp2y swp3y sw4y sw5y sw7y sw10y sw30y
%%% read the data file and get the data matrix
data = load('d_swap.txt');
%%% plot the time series
plotmargin = 50;
plot(data(:, 1), 'b'); hold on;
plot(data(:, 2), 'g'); plot(data(:, 3), 'r'); plot(data(:, 4), 'c');
plot(data(:, 5), 'm'); plot(data(:, 6), 'y'); plot(data(:, 7), 'k');
plot(data(:, 8), 'b:'); hold off; legend off;
xlim([1-plotmargin, 1257+plotmargin]);
ylim([0.9, 7.50]); grid off;
set(gca, 'Xtick', [2, 210, 419, 628, 837, 1046, 1257]);
set(gca, 'XtickLabel', ['07/05/00'; '05/03/01'; '03/11/02'; '01/09/03';...
'11/07/03'; '09/10/04'; '07/15/05']);
legend('swp1y', 'swp2y', 'swp3y', 'swp4y','swp5y', 'swp7y', ...
'swp10y', 'swp30y');
print -depsc2 dswap01
```

Example(2) Swap Rates

MATLAB Source Codes

```
%%% Take difference
ddata=diff(data);
plotmargin = 50;
plot(ddata(:, 1), 'b'); hold on;
plot(ddata(:, 2), 'g'); plot(ddata(:, 3), 'r'); plot(ddata(:, 4), 'c');
plot(ddata(:, 5), 'm'); plot(ddata(:, 6), 'y'); plot(ddata(:, 7), 'k');
plot(ddata(:, 8), 'b:'); hold off; legend off;
xlim([1-plotmargin, 1257+plotmargin]);
set(gca, 'Xtick', [1, 210, 419, 628, 837, 1046, 1256]);
set(gca, 'XtickLabel', ['07/05/00'; '05/03/01'; '03/11/02'; '01/09/03';...
'11/07/03'; '09/10/04'; '07/15/05']);
legend('swp1y', 'swp2y', 'swp3y', 'swp4y','swp5y', 'swp7y', ...
'swp10y', 'swp30y');
print -depsc2 dswap02
%%% Decomposition of Variance
[coeff, eigenvalue, explained] = pcacov(corrcoef(diff(data)));
bar(eigenvalue./sum(eigenvalue), 'y')
print -depsc2 dswap03
%%% First 3 Principal Components
mat = [1, 2, 3, 4, 5, 6, 7, 8];
plot(mat, coeff(:,1), '-', 'LineWidth', 2); hold on;
plot(mat, coeff(:,2), '-.', 'LineWidth', 2);
plot(mat, coeff(:,3), ':', 'LineWidth', 2); hold off;
ylim([-0.7, 0.65]); xlim([0.5, 8.5]);
legend('trend', 'tilt', 'convexity');
print -depsc2 dswap04
```

```
%%% reduced dimensions
pcadata = data*coeff;
plot(pcadata(:,1), '-', 'LineWidth', 1.5); hold on;
plot(pcadata(:,2), '-.', 'LineWidth', 1.5);
plot(pcadata(:,3), ':', 'LineWidth', 1.5); hold off;
xlim([-plotmargin, length(pcadata(:,1))+plotmargin]);
set(gca, 'Xtick', [1, 210, 419, 628, 837, 1046, 1256]);
set(gca, 'XtickLabel', ['07/03/00'; '05/03/01'; ...
'03/11/02'; '01/09/03';...
'11/07/03'; '09/10/04'; '07/15/05']);
legend('Level', 'Slope', 'Curvature');
print -depsc2 dswap05
%%% plot the factor loadings
bar(-coeff(:,1:3), 'grouped')
ylim([-0.7, 0.7])
legend('the 1st', 'the 2nd', 'the 3rd')
print -depsc2 dswap06
```