High Dimensional Data Analysis

Chapter 3: Principal Components Analysis

Principal Components Analysis

- Introduction
- Population Principal Components
- Sample Principal Components
- PCA and SVD
- Applications of PCA

3.1 Introduction

- Problems with high-dimensional data: Curse of dimensionality ($p \gg n$)
  - Computational time
  - Overfitting
  - Required sample size (To achieve the same accuracy, sample size needs to be n^d)
- Dimensionality reduction techniques: Aim to use fewer variables to represent the original variables while preserving as much information as possible.
- Dimensionality reduction methods include:
  - Principal Component Analysis (PCA)
  - Factor Analysis
  - Canonical Correlation Analysis
  - Multidimensional Scaling
  - Neural Networks
  - Manifold Learning
  - etc.

3.1 Introduction

- A natural approach: linearly project data into a lower-dimensional space. Let $X_{m\times n}$ be the data matrix with n observations of m dimensions. Find transformation matrix $P_{d\times m}$, where $d \leq m$, such that:

$$Y = PX$$

  This represents the original m-dimensional data points in d-dimensional space.
- PCA finds d linear combinations of the original variables that preserve most of the variance and these linear combinations are called principal components.
- PCA can be used to:
  - Detect outliers in sample points;
  - Represent original data in low-dimensional space to discover potential patterns.

Definition and Derivation of Principal Components:
Let $x_1, \cdots, x_p$ be the original variables, denote $\boldsymbol{x} = (x_1, \cdots, x_p)'$, with mean vector $\boldsymbol{\mu} = E(\mathbf{x})$ and covariance matrix $\Sigma = Var(x)$. Consider the following linear transformations:

$$\begin{cases} z_1 = a_{11}x_1 + a_{21}x_2 + \cdots + a_{p1}x_p = \boldsymbol{a}_1'\boldsymbol{x}, \\ z_2 = a_{12}x_1 + a_{22}x_2 + \cdots + a_{p2}x_p = \boldsymbol{a}_2'\boldsymbol{x}, \\ \vdots \\ z_p = a_{1p}x_1 + a_{2p}x_2 + \cdots + a_{pp}x_p = \boldsymbol{a}_p'\boldsymbol{x}, \end{cases}$$

We wish to use the composite variable z_1 to replace the original p variables $x_1, \cdots, x_p$.
$z_1, \cdots, z_p$ are the principle components and $Var(z_i) = \mathbf{a_i}'\Sigma\mathbf{a_i}$.

3.2 Population Principal Components

- **Objective of PCA**: Transform high-dimensional data into a lower-dimensional space while preserving as much variability as possible.
- Principal Components (PCs) are chosen to achieve two key properties:
  - **Maximize Variance**: Capture the directions of greatest data variability.
  - **Orthogonality**: Ensure components are uncorrelated, avoiding redundant information.
- Select unit vectors $\mathbf{a_i}$ such that:

$$Var(z_1) \geq Var(z_2) \geq \cdots \geq Var(z_p)$$

and $z_1, \ldots, z_p$ are uncorrelated.

3.2 Population Principal Components

- This can be achieved through the following iterative procedure:
  1. First principal component:

$$\hat{\mathbf{a}}_1 = \arg \max_{\|\mathbf{a}_1\|=1} \boldsymbol{a}_1' \Sigma \boldsymbol{a}_1$$

  2. Second principal component (orthogonal to the first):

$$\hat{\boldsymbol{a}}_2 = \arg \max_{\|\boldsymbol{a}_2\|=1, \boldsymbol{a}_2' \Sigma \hat{\boldsymbol{a}}_1 = 0} \boldsymbol{a}_2' \Sigma \boldsymbol{a}_2$$

  3. i-th principal component (orthogonal to all previous):

$$\hat{\boldsymbol{a}}_i = \arg \max_{\|\boldsymbol{a}_i\|=1, \boldsymbol{a}_i' \Sigma \hat{\boldsymbol{a}}_k = 0, k<i} \boldsymbol{a}_i' \Sigma \boldsymbol{a}_i$$

  4. Final principal component:

$$\hat{\boldsymbol{a}}_p = \arg \max_{\|\boldsymbol{a}_p\|=1, \boldsymbol{a}_p' \Sigma \hat{\boldsymbol{a}}_k = 0, k<p} \boldsymbol{a}_p' \Sigma \boldsymbol{a}_p$$

3.2 Population Principal Components

Solving first principal component is a conditional extremum problem, solvable using the Lagrange multiplier method. That is,

$$L(\boldsymbol{a}_1, \lambda) = \boldsymbol{a}_1' \Sigma \boldsymbol{a}_1 - \lambda(\boldsymbol{a}_1' \boldsymbol{a}_1 - 1)$$

where λ is the Lagrange multiplier.
Taking the partial derivatives with respect to $\boldsymbol{a}_1$ and λ, we obtain the following equations:

$$\begin{cases} \dfrac{\partial L(\boldsymbol{a}_1, \lambda)}{\partial \boldsymbol{a}_1} = 2(\Sigma - \lambda I_p)\boldsymbol{a}_1 = 0, \\ \dfrac{\partial L(\boldsymbol{a}_1, \lambda)}{\partial \lambda} = \boldsymbol{a}_1^{\mathrm{T}} \boldsymbol{a}_1 - 1 = 0. \end{cases}$$

Since $\boldsymbol{a}_1 \neq 0$, we have $|\Sigma - \lambda I_p| = 0$, which is essentially the problem of finding the eigenvalues and eigenvectors of Σ.

- Let $\lambda = \lambda_1$ be the largest eigenvalue of Σ, then its corresponding unit eigenvector is the desired a_1.
- Therefore, finding the i-th principal component of the original variable $\boldsymbol{x}$ is equivalent to finding the unit eigenvector $\boldsymbol{a}_i$ corresponding to the i-th largest eigenvalue λ_i of Σ.
- Furthermore, the variance of the i-th principal component z_i is

$$\text{Var}(z_i) = \boldsymbol{a}_i^T \Sigma \boldsymbol{a}_i = \lambda_i \boldsymbol{a}_i^T \boldsymbol{a}_i = \lambda_i, \quad i = 1, \cdots, p.$$

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ be the eigenvalues of Σ, and $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_p$ be the corresponding orthonormal eigenvectors, satisfying

$$A'\Sigma A = \Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{pmatrix}$$

where $A = (\boldsymbol{a}_1, \cdots, \boldsymbol{a}_p)$ is an orthogonal matrix. This decomposition is called the **spectral decomposition** of Σ.
Denote $\boldsymbol{z} = (z_1, \cdots, z_p)'$, where $z_i = \boldsymbol{a}_i' x, i = 1, \cdots, p$. The relationship between the principal components z and the original variables $\boldsymbol{x} = (x_1, \cdots, x_p)'$ is:

$$\boldsymbol{z} = A'\boldsymbol{x}, \quad \boldsymbol{x} = A\boldsymbol{z}, i.e., x_i = \sum_{k=1}^{p} a_{ik} z_k, i = 1, 2, \cdots, p.$$

Population principal components have the following properties:

Properties of Principal Components

Property 1

$Var(z) = Var(A'x) = A'\Sigma A = \Lambda$, i.e., the variances of the principal components $z_1, \cdots, z_p$ are $Var(z_i) = \lambda_i$, $i = 1, \cdots, p$, and they are uncorrelated.

Property 2 $\sum_{i=1}^{p} \sigma_{ii} = \sum_{i=1}^{p} \lambda_i$.

Proof:

$$A'\Sigma A = \Lambda \Rightarrow \Sigma = A\Lambda A' \Rightarrow tr(\Sigma) = tr(A\Lambda A') = tr(\Lambda A' A) = tr(\Lambda)$$

This property shows that the total variance of the original variables can be decomposed into the sum of the variances of the uncorrelated principal components.

3.2 Population Principal Components

Property 3 The correlation coefficient between principal component z_k and original variable x_i is

$$\rho(z_k, x_i) = \sqrt{\frac{\lambda_k}{\sigma_{ii}}} a_{ik}, \quad k, i = 1, \cdots, p,$$

where a_{ik} is the i-th element of vector $a_k = (a_{1k}, \cdots, a_{ik}, \cdots, a_{pk})'$. The correlation coefficient $\rho(z_k, x_i)$ is called the **factor loading**.

Proof: For $i, k = 1, \cdots, p$, the correlation coefficient between z_k and x_i is

$$\begin{aligned} \rho(z_k, x_i) &= \frac{\mathrm{Cov}(z_k, x_i)}{\sqrt{\mathrm{Var}(z_k)}\sqrt{\mathrm{Var}(x_i)}} \\ &= \frac{\mathrm{Cov}(z_k, \sum_{k=1}^{p} a_{ik} z_k)}{\sqrt{\mathrm{Var}(z_k)}\sqrt{\mathrm{Var}(x_i)}} = \frac{\lambda_k a_{ik}}{\sqrt{\lambda_k}\sqrt{\sigma_{ii}}} = \frac{a_{ik}\sqrt{\lambda_k}}{\sqrt{\sigma_{ii}}}. \end{aligned}$$

Property 4:

$$\sum_{k=1}^{p} \rho^2(z_k, x_i) = \sum_{k=1}^{p} \frac{\lambda_k a_{ik}^2}{\sigma_{ii}} = 1, \quad i = 1, 2, \cdots, p.$$

Proof: Since $\Sigma = A\Lambda A^T$, we have

$$\sigma_{ii} = (a_{1k}, \cdots, a_{pk})\Lambda \begin{pmatrix} a_{1k} \\ \vdots \\ a_{pk} \end{pmatrix} = \sum_{k=1}^{p} \lambda_k a_{ik}^2$$

then

$$\sum_{k=1}^{p} \rho^2(z_k, x_i) = \sum_{k=1}^{p} \frac{a_{ik}^2 \lambda_k}{\sigma_{ii}} = 1, \quad i = 1, \cdots, p.$$

Property 5:

$$\sum_{i=1}^{p} \sigma_{ii} \rho^2(z_k, x_i) = \lambda_k, \quad k = 1, 2, \cdots, p.$$

Proof: According to **Property 3** and $\boldsymbol{a}_k^T \boldsymbol{a}_k = 1$, we have

$$\sum_{i=1}^{p} \sigma_{ii} \rho^2(z_k, x_i) = \sum_{i=1}^{p} \sigma_{ii} \frac{\lambda_k a_{ik}^2}{\sigma_{ii}} = \lambda_k \sum_{i=1}^{p} a_{ik}^2 = \lambda_k$$

3.2 Population Principal Components

cumulative contribution rate

One purpose of PCA is dimensionality reduction, i.e., reducing the number of variables. Therefore, in practice, we usually do not use all p principal components but select the first $m(m < p)$ principal components with larger variances. The appropriate value of m can be determined based on the contribution rate of the principal components.

Definition The proportion of the variance of principal component z_k to the total variance, $\frac{\lambda_k}{\sum_{i=1}^{p} \lambda_i}$, is called the **contribution rate** of principal component z_k. The sum of the contribution rates of the first m principal components, $\sum_{k=1}^{m} \lambda_k / \sum_{i=1}^{p} \lambda_i$, is called the **cumulative contribution rate** of $z_1, \cdots, z_m$.

Usually, we choose the smallest possible m such that the cumulative contribution rate of $z_1, \cdots, z_m$ reaches a relatively high percentage, e.g., above 75%.

Definition The sum of squared correlation coefficients between the original variable x_i and the principal components $z_1, \cdots, z_m$ is called the contribution rate of the first m principal components to the original variable x_i, denoted $v_i^{(m)}$, i.e.,

$$v_i^{(m)} = \sum_{k=1}^{m} \rho^2(z_k, x_i) = \sum_{k=1}^{m} \frac{\lambda_k a_{ik}^2}{\sigma_{ii}}$$

Example 1 Let the covariance matrix of random vector $x = (x_1, x_2, x_3)'$ be

$$\Sigma = \begin{pmatrix} 1 & -2 & 0 \\ -2 & 5 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

Find the principal components z_1, z_2, z_3 of x and their contribution rates.
Solution: After calculation, the eigenvalues of Σ are $\lambda_1 = 3 + \sqrt{8}, \lambda_2 = 2, \lambda_3 = 3 - \sqrt{8}$, and the corresponding orthonormal eigenvectors are

$$a_1 = \begin{pmatrix} 0.383 \\ -0.924 \\ 0.000 \end{pmatrix}, a_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, a_3 = \begin{pmatrix} 0.924 \\ 0.383 \\ 0.000 \end{pmatrix}.$$

Therefore, the principal components are:

$$z_1 = 0.383x_1 - 0.924x_2, \ z_2 = x_3, \ z_3 = 0.924x_1 + 0.383x_2.$$

When $m = 1$, the contribution rate of z_1 reaches 72.85%. When $m = 2$, the cumulative contribution rate of z_1, z_2 reaches 97.85%.

3.3 Sample Principal Components

In practice, Σ and R are often unknown and need to be estimated from samples. Let the sample data matrix be

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} x'_{(1)} \\ \vdots \\ x'_{(n)} \end{pmatrix},$$

The sample mean vector is $\bar{x} = (\bar{x}_1, \cdots, \bar{x}_p)^T$. The sample covariance matrix and sample correlation matrix are:

$$S = (s_{ij})_{p\times p}, \quad R = (r_{ij})_{p\times p},$$

where

$$s_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j), \quad r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii}s_{jj}}}, \quad i, j = 1, \cdots, p.$$

3.3 Sample Principal Components

We can use the sample covariance matrix S as an estimate of the population covariance matrix Σ, or use the sample correlation matrix R as an estimate of the population correlation matrix, and then obtain the sample principal components using the method from the previous section.

Obtaining Principal Components from the Sample Correlation Matrix:
Let $\lambda_1^* \geq \lambda_2^* \geq \cdots \geq \lambda_p^* \geq 0$ be the eigenvalues of the sample correlation matrix R, and $u_1^*, u_2^*, \cdots, u_p^*$ be the corresponding orthonormal eigenvectors. Then the **i-th sample principal component** is

$$z_i^* = u_i^{*'} x^*, \quad i = 1, \cdots, p,$$

where $x^* = (x_1^*, \cdots, x_p^*)'$ is the vector of original variables $x = (x_1, \cdots, x_p)$ standardized.

3.3 Sample Principal Components

Let

$$z^* = (z_1^*, \cdots, z_p^*)' = U^{*'} x^*,$$

where $U^* = (u_1^*, \cdots, u_p^*) = (u_{ij}^*)_{p\times p}$. Comparing with the properties of population principal components, the sample principal components have the following properties.

1. $Var(z_k^*) \approx u_k^{*'} R u_k^* = \lambda_k^*, \quad k = 1, \cdots, p.$
2. $Cov(z_k^*, z_j^*) \approx u_k^{*'} R u_j^* = 0, \quad k \neq j, \quad k, j = 1, \cdots, p.$
3. $\sum_{j=1}^p \lambda_j^* = tr(R) = p.$
4. The correlation coefficient between sample principal component z_k^* and standardized original variable x_i^* is

$$\rho(z_k^*, x_i^*) \approx \sqrt{\lambda_k^*} u_{ik}^*, \quad k, i = 1, \cdots, p.$$

3.3 Sample Principal Components

Let

$$z^*_{(t)} = (z^*_{t1}, \cdots, z^*_{tp})' = U^{*'} x^*_{(t)}, \quad t = 1, \cdots, n,$$

then the principal component score matrix corresponding to the standardized data matrix $X^* = (x^*_{ij})_{n\times p}$ is

$$Z^* = (z^*_{ij})_{n\times p} = \begin{pmatrix} z^*_{11} & z^*_{12} & \cdots & z^*_{1p} \\ z^*_{21} & z^*_{22} & \cdots & z^*_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ z^*_{n1} & z^*_{n2} & \cdots & z^*_{np} \end{pmatrix} = \begin{pmatrix} z^{*'}_{(1)} \\ z^{*'}_{(2)} \\ \vdots \\ z^{*'}_{(n)} \end{pmatrix} = X^* U^*.$$

The first column of this score matrix contains the scores of the first principal component for the n samples, the second column contains the scores of the second principal component, and so on. Using the first principal component scores or the comprehensive scores of the first m principal components, samples can be sorted or evaluated.

Interpretation of Principal Components:
The interpretation of principal components is related to the practical background of the problem being analyzed. Giving a reasonable explanation for the specific meaning of principal components based on their loadings is an important aspect of PCA.

One purpose of PCA is to simplify the data structure for easier computation and analysis, but the ultimate goal is to solve problems.

For some practical problems, the effectiveness of PCA depends on whether a good explanation can be given for the practical meaning of the selected principal components.

3.3 Sample Principal Components

Example 2 Randomly select 30 students from a certain grade in a middle school, measure their 4 body indicators: height (x_1), weight (x_2), chest circumference (x_3), and sitting height (x_4). The data are shown in Table 3.1. Perform PCA on the body indicator data of these 30 students.
Solution Calculate the correlation matrix R from the data in Table 3.1 and perform PCA based on this matrix. The calculated eigenvalues, eigenvectors (loading vectors), and contribution rates are listed in Table 3.2.

Table 3.1. Data of four body indicators for 30 middle school students

Serial No.	x_1	x_2	x_3	x_4	Serial No.	x_1	x_2	x_3	x_4
1	148	41	72	78	16	152	35	73	79
2	139	34	71	76	17	149	47	82	79
3	160	49	77	86	18	145	35	70	77
4	149	36	67	79	19	160	47	74	87
5	159	45	80	86	20	156	44	78	85
6	142	31	66	76	21	151	42	73	82
7	153	43	76	83	22	147	38	73	78
8	150	43	77	79	23	157	39	68	80
9	151	42	77	80	24	147	30	65	75
10	139	31	68	74	25	157	48	80	88
11	140	29	64	74	26	151	36	74	80
12	161	47	78	84	27	144	36	68	76
13	158	49	78	83	28	141	30	67	76
14	140	33	67	77	29	139	32	68	73
15	137	31	66	73	30	148	38	70	78

Variable	Eigenvectors u_i^*			
	PC1	PC2	PC3	PC4
x_1^*	0.497	0.543	-0.450	0.506
x_2^*	0.515	-0.210	-0.462	-0.691
x_3^*	0.481	-0.725	0.175	0.461
x_4^*	0.507	0.368	0.744	-0.232
Eigenvalue $\sqrt{\lambda_i}$	1.8817805	0.55980636	0.28179594	0.25711844
Contribution rate	0.8852745	0.07834579	0.01985224	0.01652747
Cumulative	0.8852745	0.96362029	0.98347253	1.00000000

Table: Eigenvalues, eigenvectors, contribution rates, and cumulative contribution rates

3.3 Sample Principal Components

From the contribution rates listed in Table 3.2, the contribution rate of the first principal component is close to 88.53%, and the cumulative contribution rate of the first two principal components reaches 96.36%. Therefore, selecting only the first two principal components can well summarize this data set. These two principal components are:

$$z_1^* = 0.497x_1^* + 0.515x_2^* + 0.481x_3^* + 0.507x_4^*,$$

$$z_2^* = 0.543x_1^* - 0.210x_2^* - 0.725x_3^* + 0.368x_4^*.$$

It can be seen that the first and second principal components are linear combinations of the standardized variables, and their coefficients are the components of the corresponding eigenvectors.

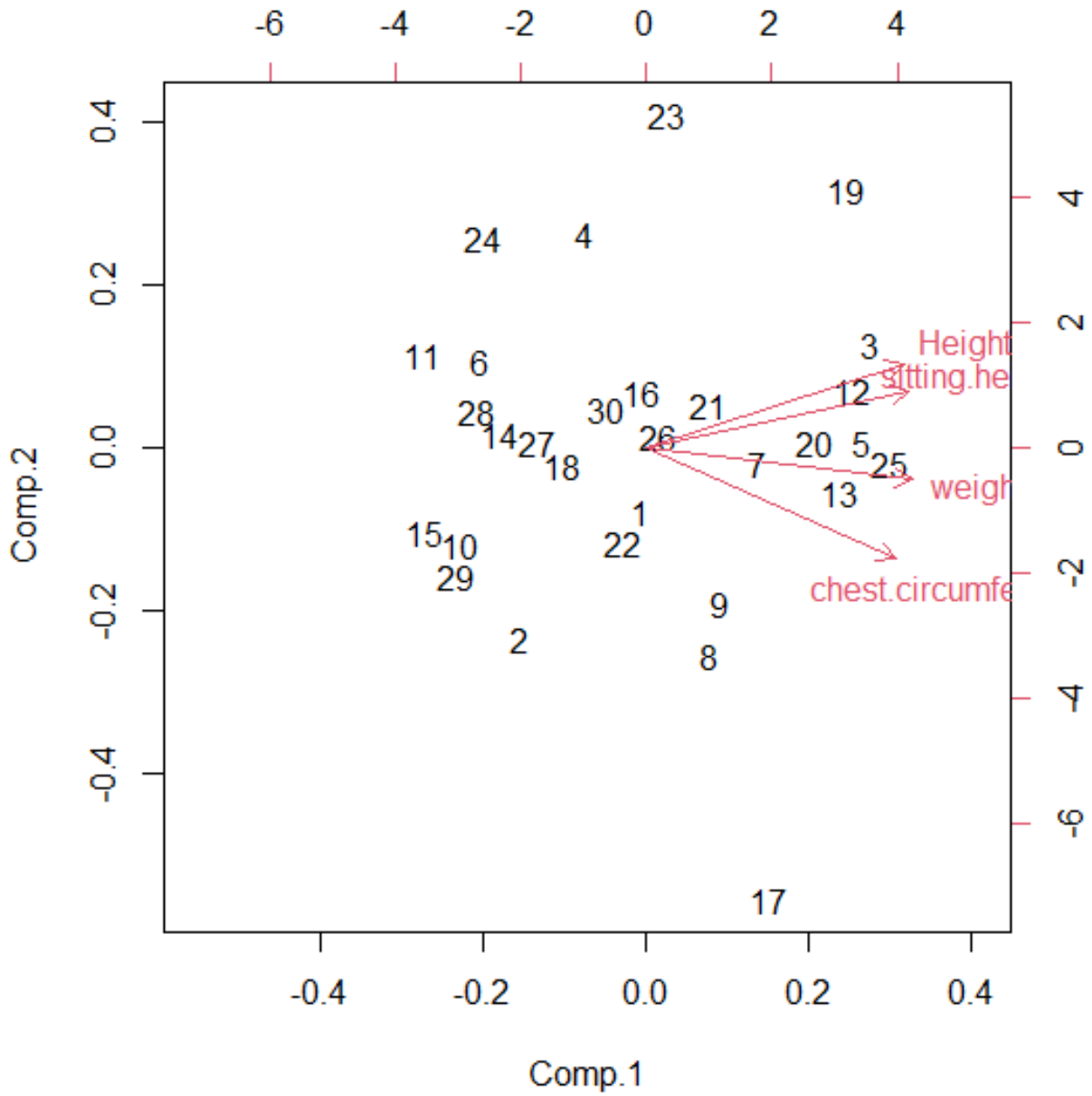


Figure 3.1 The scatter plot of the first and second PCs for the body

3.3 Sample Principal Components

- The actual meaning of each principal component can be interpreted using the values of the eigenvector components (loadings) corresponding to the variables.
- In this example, all components of the eigenvector corresponding to the largest eigenvalue are positive and around 0.5. It reflects the stoutness of the middle school students' physique.
- Students with tall and large builds have larger values for all four indicators, so the value of the first principal component is larger; students with short and small builds have smaller values for all four indicators, so the value of the first principal component is smaller.
- Therefore, the first principal component is the "physique size" component. The second principal component is the difference between height and breadth. Students with large second principal component values have slender builds, while those with small values have stocky builds. Therefore, the second principal component is the "body type" component.

3.3 Sample Principal Components

- It is important to note that the signs of the components of the principal component loading matrix's column vectors can sometimes change. This is because the column vectors of the principal component loadings are the eigenvectors of the sample covariance matrix or sample correlation matrix, and the components of an eigenvector can all be multiplied by -1. Different software, or two calculations by the same software, might yield eigenvectors differing by a sign.
- Therefore, the column vectors of the principal component loadings might differ by a sign in different calculations; in other words, the direction of the loadings might be reversed.

Python Code for Multiple Correlation

```
import pandas as pd
import numpy as np
import statsmodels.api as sm

# Create DataFrame from data
data = {'ID': [1, 2, 3, ...], 'x1': [44, 40, 44, ...],
        'x2': [89.47, 75.07, 85.84, ...], 'x3': [44.609, 45.313, 54.297, ...],
        'x4': [11.37, 10.07, 8.65, ...], 'x5': [62, 62, 45, ...],
        'x6': [178, 185, 156, ...], 'x7': [182, 185, 168, ...]}
df = pd.DataFrame(data)

# Predictors and dependent variable
X = df[['x1', 'x2', 'x4', 'x5', 'x6', 'x7']]
y = df['x3']

# Add constant and fit OLS model
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()

# Calculate multiple correlation coefficient
R = np.sqrt(model.rsquared)
print(f"R_{{3·1,2,4,5,6,7}} = {R:.4f}")
```

3.4 PCA and Singular Value Decomposition (SVD)

- Let the data matrix be $X_{n\times p}$, where n is the number of samples and p is the number of variables/dimensions
- According to Singular Value Decomposition (SVD), there exist orthogonal matrices $U_{n\times n}$ and $V_{p\times p}$, and an $n \times p$ diagonal matrix Λ, such that:

$$X = U\Lambda V'$$

- Multiply by X' to obtain:

$$X'X = (U\Lambda V')'U\Lambda V' = V\Lambda'\Lambda V'$$
$$XX' = (U\Lambda V')(U\Lambda V')' = U\Lambda\Lambda' U'$$

- $X'X$ and XX' share the same non-zero eigenvalues
- If u is an eigenvector of XX' corresponding to eigenvalue w, then...

3.4 PCA and SVD (Continued)

$$XX'u = wu \Longrightarrow X'(XX'u) = X'X(X'u) = w(X'u)$$

This means $X'u$ is the eigenvector of $X'X$.

PCA by SVD

- Let M be the centered $n \times p$ data matrix, then the sample covariance matrix is:
$$S = \frac{1}{n-1} M'M$$
- From SVD: $M = U\Lambda V'$, therefore:
$$S = \frac{1}{n-1} V\Lambda'\Lambda V'$$
- Thus, V contains the principal component loadings, and $MV = Y$ gives the principal component scores.
- Taking the first k principal components:
$$M = MVV' = M(V_k V_k' + V_{p-k} V_{p-k}') \approx (MV_k)V_k' = \hat{Y}_k V_k' = \hat{M}$$

Classification of Indicators

The following example illustrates how to use PCA to classify indicators.

Example 3 Clothing sizing classification problem. Measure 16 indicators for 128 adult men: height (x1), sitting height (x2), chest circumference (x3), head height (x4), trouser length (x5), inseam (x6), hand length (x7), neck circumference (x8), front chest (x9), back (x10), shoulder thickness (x11), shoulder width (x12), sleeve length (x13), waist circumference (x14), hip circumference (x15), and calf circumference (x16). The sample correlation matrix for the 16 indicators is given in Table 7.6 (only upper triangle shown). Perform PCA based on the correlation matrix and classify the 16 indicators.

Table 3.3 Correlation matrix of 16 body indicator data

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_1	1.0	0.79	0.3	0.9	0.8	0.7	0.7	0.2	0.21	0.26	0.07	0.52	0.77	0.25	0.51	0.21
x_2		1.0	0.3	0.7	0.5	0.5	0.5	0.1	0.07	0.16	0.21	0.41	0.47	0.17	0.35	0.16
x_3			1.0	0.3	0.3	0.3	0.3	0.5	0.5	0.28	0.33	0.38	0.39	0.41	0.64	0.58
x_4				1.0	0.9	0.7	0.7	0.2	0.20	0.22	0.08	0.53	0.79	0.27	0.57	0.26
x_5					1.0	0.7	0.7	0.2	0.18	0.23	-0.02	0.48	0.79	0.27	0.51	0.23
x_6						1.0	0.7	0.1	0.18	0.23	0.00	0.38	0.69	0.14	0.26	0.00
x_7							1.0	0.2	0.29	0.25	0.10	0.44	0.67	0.16	0.38	0.12
x_8								1.0	-0.04	0.49	0.44	0.30	0.32	0.51	0.51	0.38
x_9									1.0	-0.34	-0.16	-0.05	0.23	0.21	0.15	0.18
x_{10}										1.0	0.23	0.50	0.31	0.15	0.29	0.14
x_{11}											1.0	0.24	0.10	0.31	0.28	0.31
x_{12}												1.0	0.62	0.17	0.41	0.18
x_{13}													1.0	0.26	0.50	0.24
x_{14}														1.0	0.63	0.50
x_{15}															1.0	0.65
x_{16}																1.0

3.5 Applications of PCA

In this example, $p = 16, n = 128$. PCA from the correlation matrix yields the first three eigenvalues and contribution rates shown in Table 3.4.

Table 3.4. First 3 eigenvalues and contribution rates for Example 3.3

	PC1	**PC2**	**PC3**
Eigenvalue λ_k^*	7.036	2.614	1.632
Contribution Rate	0.440	0.163	0.102
Cumulative Rate	0.440	0.603	0.705

The cumulative contribution rate of the first three PCs is 70.5%. Selecting the first three PCs basically summarizes this data.

3.5 Applications of PCA

The loading vectors (eigenvectors) for the first three PCs are:

$$
\begin{aligned}
u_1' &= (0.342, 0.265, 0.234, 0.344, 0.326, 0.286, 0.295, 0.189, 0.000, \\
&\quad 0.154, 0.000, 0.243, 0.317, 0.180, 0.266, 0.158)' \\
u_2' &= (0.200, 0.143, -0.329, 0.181, 0.200, 0.270, 0.192, -0.370, 0.000, \\
&\quad -0.174, -0.348, 0.000, 0.112, -0.371, -0.271, -0.363)' \\
u_3' &= (0.000, 0.000, 0.140, 0.000, 0.000, 0.000, 0.000, -0.150, 0.626, \\
&\quad -0.528, -0.202, -0.315, 0.000, 0.252, 0.135, 0.243)'
\end{aligned}
$$

- PC1 loadings are all positive, larger loads for height (x1), sitting height (x2), head height (x4), trouser length (x5), inseam (x6), hand length (x7), sleeve length (x13): "**Length** factor".
- PC2 has large negative loads for chest (x3), neck (x8), shoulder thick (x11), waist (x14), hip (x15), calf (x16): "**Girth** factor".
- PC3 has large loads for front chest (x9), back (x10), shoulder width (x12): "**Body Type** factor".

The following Figure 3.2 shows the scatter plot of the loadings of the first two principal components. The points in the upper right corner of the plot can be considered as one category, which are the "**length**" indicators; the points in the lower left corner can be considered as another category, which are the "**girth**" indicators; the points in the middle are one category, which are the "**body shape**" indicators.
Based on the above analysis, the 16 indicators can be classified as follows:

- First category: the "**length**" indicators: height, sitting height, head height, pants length, crotch height, hand length, and sleeve length;
- Second category: the "**girth**" indicators: chest circumference, collar circumference, shoulder thickness, rib circumference, waist circumference, and calf circumference;
- Third category: the "**body shape**" indicators: chest width, back width, and shoulder width.

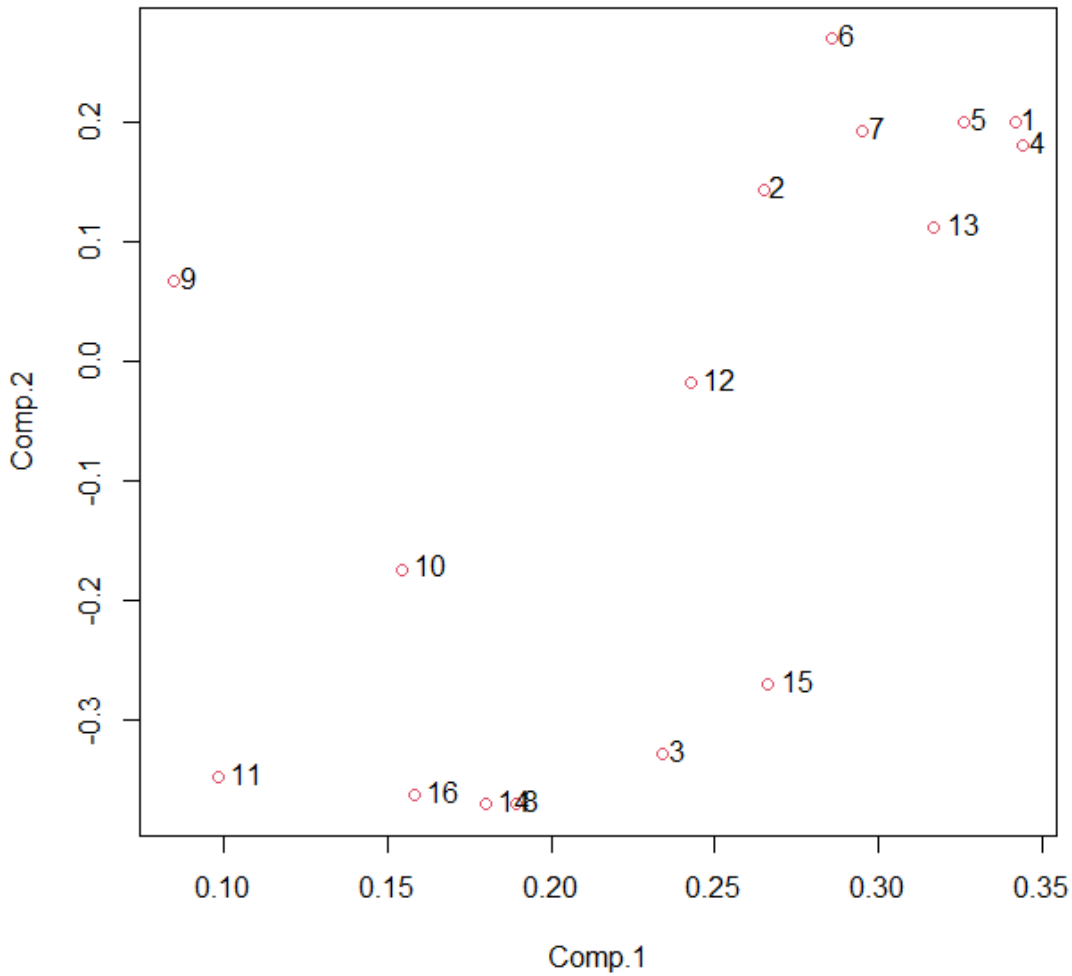


Figure 3.2. The scatter plot of the loadings of the first two principal components.

3.5 Applications of PCA

Classification and Sorting of Samples

- In practice, we often encounter the problem of sorting or evaluating samples in multi-indicator systems. PCA is a common method for this.
- Generally, the scores of the **first principal component** can be used to sort the samples.
- Some literature proposes using a **weighted average** of the principal component scores for sorting. This method calculates the weighted average of the scores of each principal component. However, before weighting, the direction of each principal component needs to be adjusted consistently (e.g., so that the highest score corresponds to the "best" sample).
- It can be proven that the information content of this weighted average method is less than that of using only the first principal component, but its direction is a synthesis of the directions of the various principal components, which might be the rationality of the comprehensive sorting method.

Python Code for Example 3

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load data
df = pd.read_csv('LA_Neighborhoods.txt')

# Standardize data
scaler = StandardScaler()
scaled_df = scaler.fit_transform(df)

# Perform PCA
pca = PCA()
pca.fit(scaled_df)

# Eigenvalues
eig_vals = pca.explained_variance_

# Contribution rates
con_rates = pca.explained_variance_ratio_

# Cumulative contribution rates
cum_rates = np.cumsum(con_rates)

# Print results
print("Eigenvalues:", eig_vals)
print("Contribution Rates:", con_rates)
print("Cumulative Rates:", cum_rates)

# Scree Plot
plt.figure(figsize=(8, 6))
plt.plot(range(1, len(eig_vals) + 1), con_rates, 'bo-')
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Contribution Rate')
plt.grid(True)
plt.show()
```

3.5 Applications of PCA

Example 4 Perform PCA on the Los Angeles neighborhoods data.

Variable Name	Description	Type
LA.Nbhd	Neighborhood Name	Categorical
Income	Median Income	Quantitative
Schools	Public School API Score Median	Quantitative
Diversity	Ethnic Diversity (0-10)	Categorical/Ordinal
Age	Median Age	Quantitative
Homes	Homeownership Rate	Quantitative
Vets	Veteran Percentage	Quantitative
Asian	Asian Percentage	Quantitative
Black	Black Percentage	Quantitative
Latino	Latino Percentage	Quantitative
White	White Percentage	Quantitative
Population	Population	Quantitative
Area	Area	Quantitative
Longitude	Longitude	Quantitative
Latitude	Latitude	Quantitative

Note: API stands for Academic Performance Index. Data available from:
http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_LA_Neighborhoods_Data

Principal Component Analysis

Remove population and area, add population density variable

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler

# Read data
w = pd.read_csv("D://data/LA.Neighborhoods.csv")

# Add population density variable
w['density'] = w['Population'] / w['Area']

# Remove population, area, longitude, latitude variables
u = w.drop(w.columns[11:15], axis=1)

# Remove the first column (neighborhood names) for PCA calculation
u_numeric = u.iloc[:, 1:]
```

Principal Component Analysis

```
# Standardize the data
scaler = StandardScaler()
u_scaled = scaler.fit_transform(u_numeric)

# Calculate correlation matrix and eigen decomposition
corr_matrix = np.corrcoef(u_scaled.T)
eigenvalues, eigenvectors = np.linalg.eig(corr_matrix)

# Sort eigenvalues and eigenvectors in descending order
idx = eigenvalues.argsort()[::-1]
eigenvalues = eigenvalues[idx]
eigenvectors = eigenvectors[:, idx]

# Calculate contribution rate and cumulative contribution rate
contribution_rate = eigenvalues / np.sum(eigenvalues)
cumulative_contribution_rate = np.cumsum(contribution_rate)
```

PCA: Eigenvalues and Contribution Rates

```
# Print results
print("Eigenvalues:", eigenvalues)
print("Contribution Rate:", contribution_rate)
print("Cumulative Contribution Rate:",cumulative_contribution_rate)
```

Table 3.5. Eigenvalues, contribution rates, and cumulative contribution rates

Component	Eigenvalue	Contribution Rate	Cumulative Rate
1	4.6930	0.4266	0.4266
2	1.8176	0.1652	0.5919
3	1.1688	0.1063	0.6981
4	1.0270	0.0934	0.7915
5	0.8063	0.0733	0.8648
6	0.5577	0.0507	0.9155
7	0.4099	0.0373	0.9528
8	0.2437	0.0222	0.9749
9	0.1871	0.0170	0.9919
10	0.0883	0.0080	0.9999
11	0.0006	0.0001	1.0000

```
# Plot scree plot and cumulative contribution plot
plt.figure(figsize=(12, 5))
# Scree plot
plt.subplot(1, 2, 1)
plt.plot(range(1, len(eigenvalues) + 1), eigenvalues, 'o-',
      markersize=8, linewidth=2)
plt.title('Scree Plot')
plt.xlabel('Component Number')
plt.ylabel('Eigen Value')
plt.grid(True)
# Cumulative contribution plot
plt.subplot(1, 2, 2)
plt.plot(range(1, len(cumulative_contribution_rate) + 1),
     cumulative_contribution_rate, 'o-', markersize=8, linewidth=2)
plt.title('Cumulative Contribution')
plt.xlabel('Component Number')
plt.ylabel('Cumulative Contribution')
plt.grid(True)
plt.tight_layout()
plt.show()
```

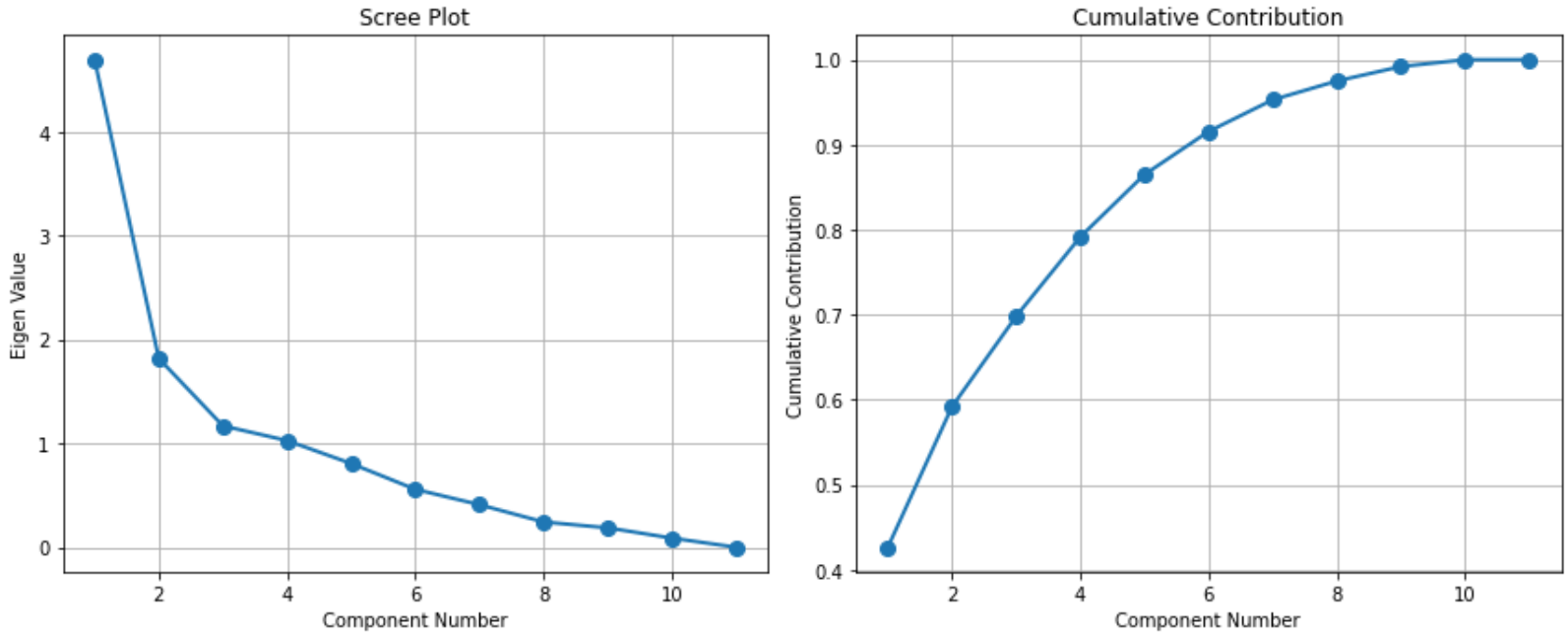


Figure 3.3. Scree Plot and Cumulative Contribution Plot

Solution

- First, calculate the eigenvalues, contribution rates, and cumulative contribution rates of the sample correlation matrix. The results are listed in Table 3.5.
- Figure 3.3 is a plot based on this data. The left plot is the contribution rate plot, called the **Scree Plot**. The name comes from the hope that the graph is steep initially, like a cliff, and the remaining values are small, like rubble at the bottom.
- From Table 3.5 and Figure 3.3, the first PC has a large contribution rate, while others are smaller. The cumulative contribution rate exceeds 79% only after the fourth PC.

PC Loadings Calculation

```
# Calculate loadings (correlations between
# first four PCs and original variables)
# loadings = eigenvectors * sqrt(eigenvalues) for each component
loadings = eigenvectors[:, :4] * np.sqrt(eigenvalues[:4])

# Convert to DataFrame for better labeling
loadings_df = pd.DataFrame(loadings,
                           index=u_numeric.columns,
                           columns=[f'PC{i+1}' for i in range(4)])

print("Loadings (PCs vs Original Variables):")
print(loadings_df)
```

PCA: Correlations between PCs and Original Variables

Table 3.6. Loadings (PCs vs Original Variables)

Original Variable	PC1	PC2	PC3	PC4
Income	0.8852	-0.0656	-0.2212	-0.0301
Schools	0.0662	0.3459	-0.1050	0.7980
Diversity	-0.0992	0.6770	0.6011	-0.0059
Age	0.9247	0.0488	0.1288	-0.1779
Homes	0.6853	0.0650	0.1261	0.3255
Vets	0.7957	-0.2809	0.3605	0.1356
Asian	0.0526	0.8583	0.1721	-0.1944
Black	-0.3156	-0.6302	0.6497	0.0870
Latino	-0.8319	0.0535	-0.2478	0.2261
White	0.8751	0.0900	-0.2673	-0.1865
density	-0.6136	0.0685	0.0101	-0.3190

Table 3.6 shows the coefficient vectors (loadings) for the first four principal components, where blank spaces indicate loadings close to 0.

- PC1 has large positive loadings for high income, older age, high home ownership, many veterans, and high European ancestry: "**Socioeconomic Status**" component.
- PC2 has negative loadings for high Latino population, high population density.
- PC3 has large loadings for high African American population; negative loads for high Asian population, high racial diversity: "**Ethnicity**" component.
- PC4 has negative loads for many public schools: "**School District**" component.

PC Loadings Plot

```
# Create the loadings plots
plt.figure(figsize=(12, 5))

# Plot for PC1 vs PC2
plt.subplot(1, 2, 1)
plt.xlim(-1.2, 1.2)
plt.ylim(-1.2, 1.2)
plt.axhline(y=0, color='k', linestyle='-', linewidth=0.5)
plt.axvline(x=0, color='k', linestyle='-', linewidth=0.5)
plt.xlabel('Component 1 (42%)')
plt.ylabel('Component 2 (17%)')
plt.title('Loadings Plot: PC1 vs PC2')

# Add variable names as text
for i, var_name in enumerate(u_numeric.columns):
    plt.text(loadings[i, 0], loadings[i, 1], var_name, fontsize=9)
```

PC Loadings Plot

```
# Plot for PC3 vs PC4
plt.subplot(1, 2, 2)
plt.xlim(-1.2, 1.2)
plt.ylim(-1.2, 1.2)
plt.axhline(y=0, color='k', linestyle='-', linewidth=0.5)
plt.axvline(x=0, color='k', linestyle='-', linewidth=0.5)
plt.xlabel('Component 3 (11%)')
plt.ylabel('Component 4 (9%)')
plt.title('Loadings Plot: PC3 vs PC4')

# Add variable names as text
for i, var_name in enumerate(u_numeric.columns):
    plt.text(loadings[i, 2], loadings[i, 3], var_name, fontsize=9)

plt.tight_layout()
plt.show()
```

PC Loadings Plot

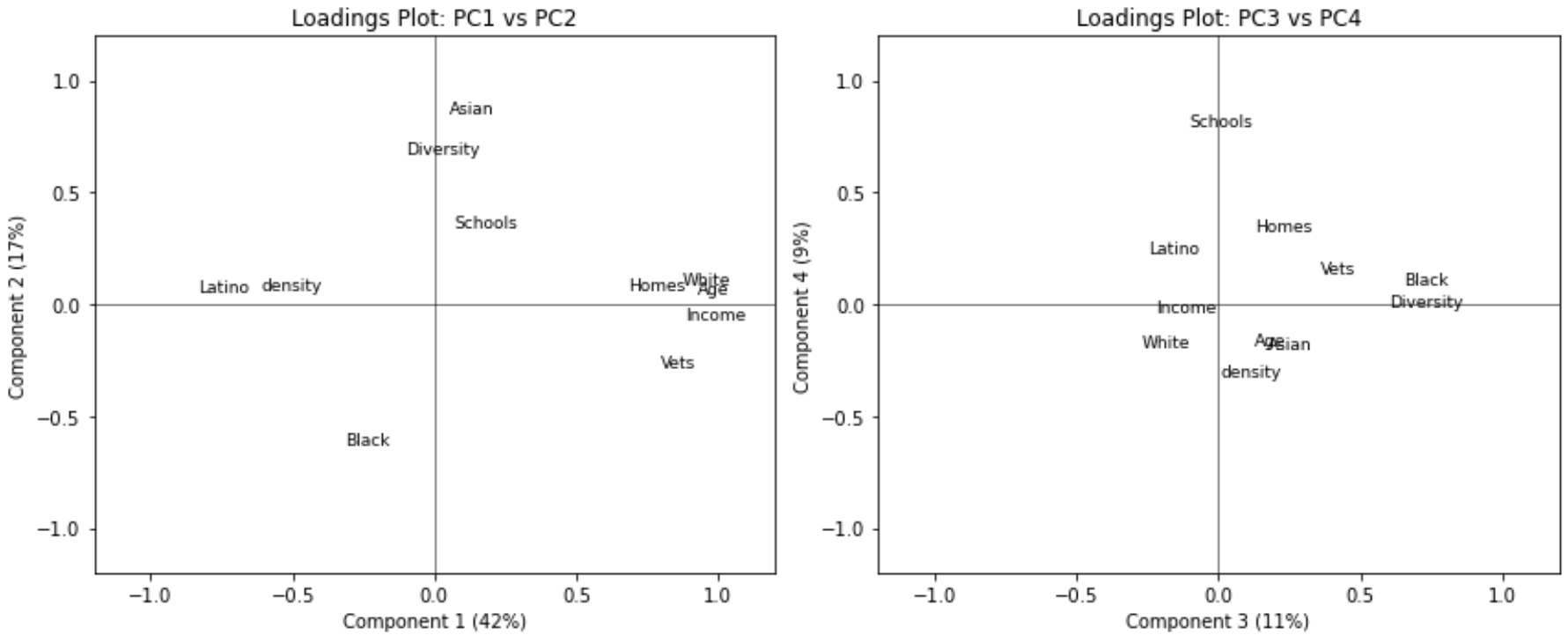


Figure 3.4. PCA Loadings Plot

Interpretation

PC1 & PC2 (Left of Figure 3.4)

- Large PC1 corresponds to communities with:
  - Higher income
  - Older population
  - More homeowners
  - More veterans
  - Higher White population
- Small PC1 corresponds to communities with:
  - Higher Latino population
  - Higher population density
- Large PC2 corresponds to communities with:
  - Higher Asian population
  - ethnic diversity
- Small PC2 corresponds to communities with more Black population
- Other variables show weak correlation with these two principal components

PC3 & PC4 (Right of Figure 3.4)

- Points are closer to the center (correlations near 0) compared to left figure
- Large PC3 corresponds to communities with:
  - Higher Black population
  - Higher racial diversity
- Large PC4 corresponds to communities with better school performance

PC Scores Calculation and Plot

```
# Calculate principal component scores
scores = np.dot(u_scaled, eigenvectors[:, :4])
neighborhood_names = u.iloc[:, 0]
# Create figure with two subplots
plt.figure(figsize=(12, 5))

# First subplot: PC1 vs PC2
plt.subplot(1, 2, 1)
plt.xlim(-7, 8)
plt.ylim(-6, 6)
plt.axhline(y=0, color='red', linestyle='-', linewidth=0.5)
plt.axvline(x=0, color='red', linestyle='-', linewidth=0.5)
plt.xlabel('Component 1')
plt.ylabel('Component 2')
plt.title('Sample Principal Components')

# Add neighborhood names as text labels
for i, neighborhood in enumerate(neighborhood_names):
    plt.text(scores[i, 0], scores[i, 1], neighborhood, fontsize=6)
```

PC Scores Calculation and Plot

```
# Second subplot: PC3 vs PC4
plt.subplot(1, 2, 2)
plt.xlim(-7, 6)
plt.ylim(-6, 6)
plt.axhline(y=0, color='red', linestyle='-', linewidth=0.5)
plt.axvline(x=0, color='red', linestyle='-', linewidth=0.5)
plt.xlabel('Component 3')
plt.ylabel('Component 4')
plt.title('Sample Principal Components')

# Add neighborhood names as text labels
for i, neighborhood in enumerate(neighborhood_names):
    plt.text(scores[i, 2], scores[i, 3], neighborhood, fontsize=6)

plt.tight_layout()
plt.show()
```

PC Scores Plot

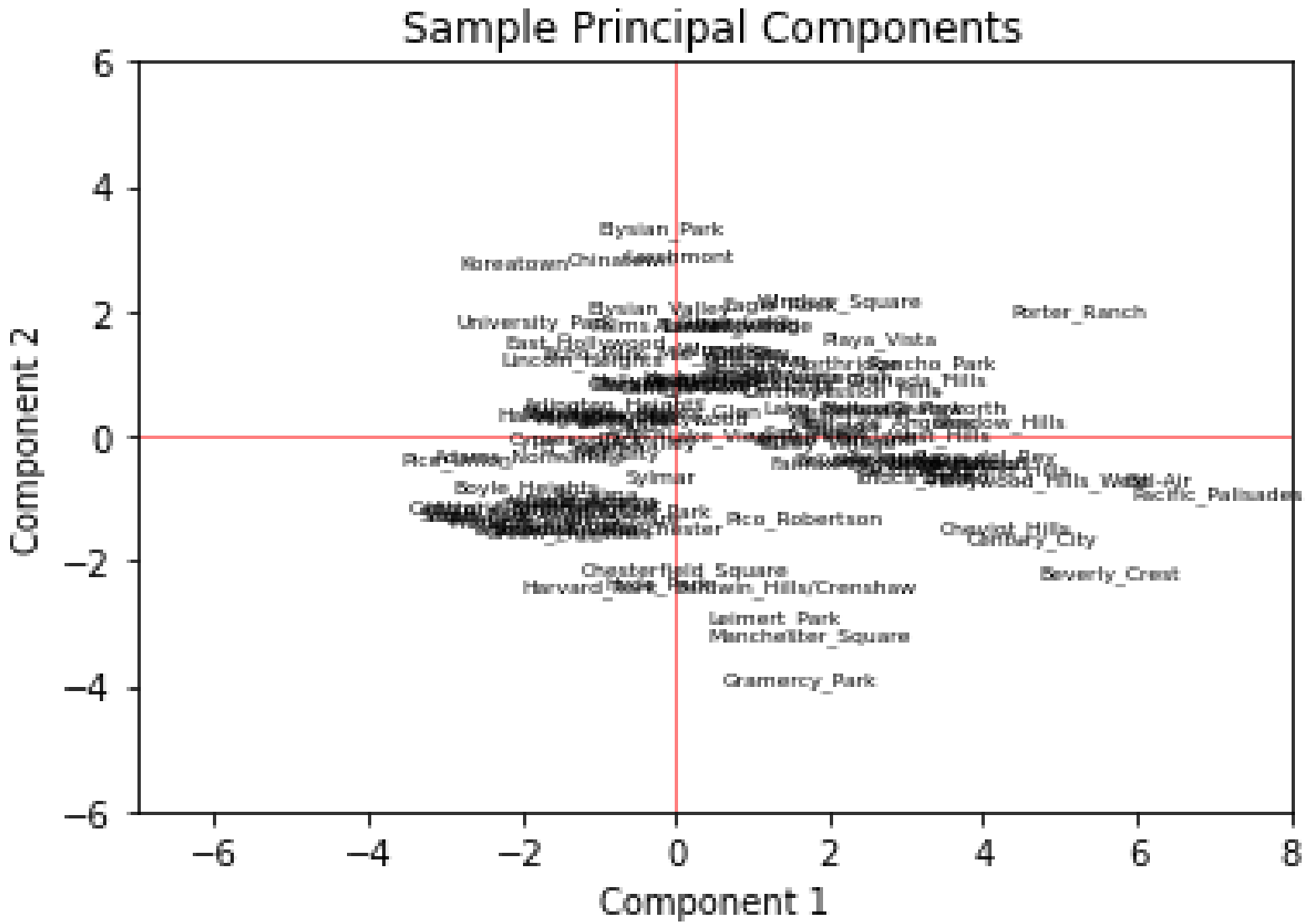


Figure 3.5.1 PC Scores Plot

PC Scores Plot

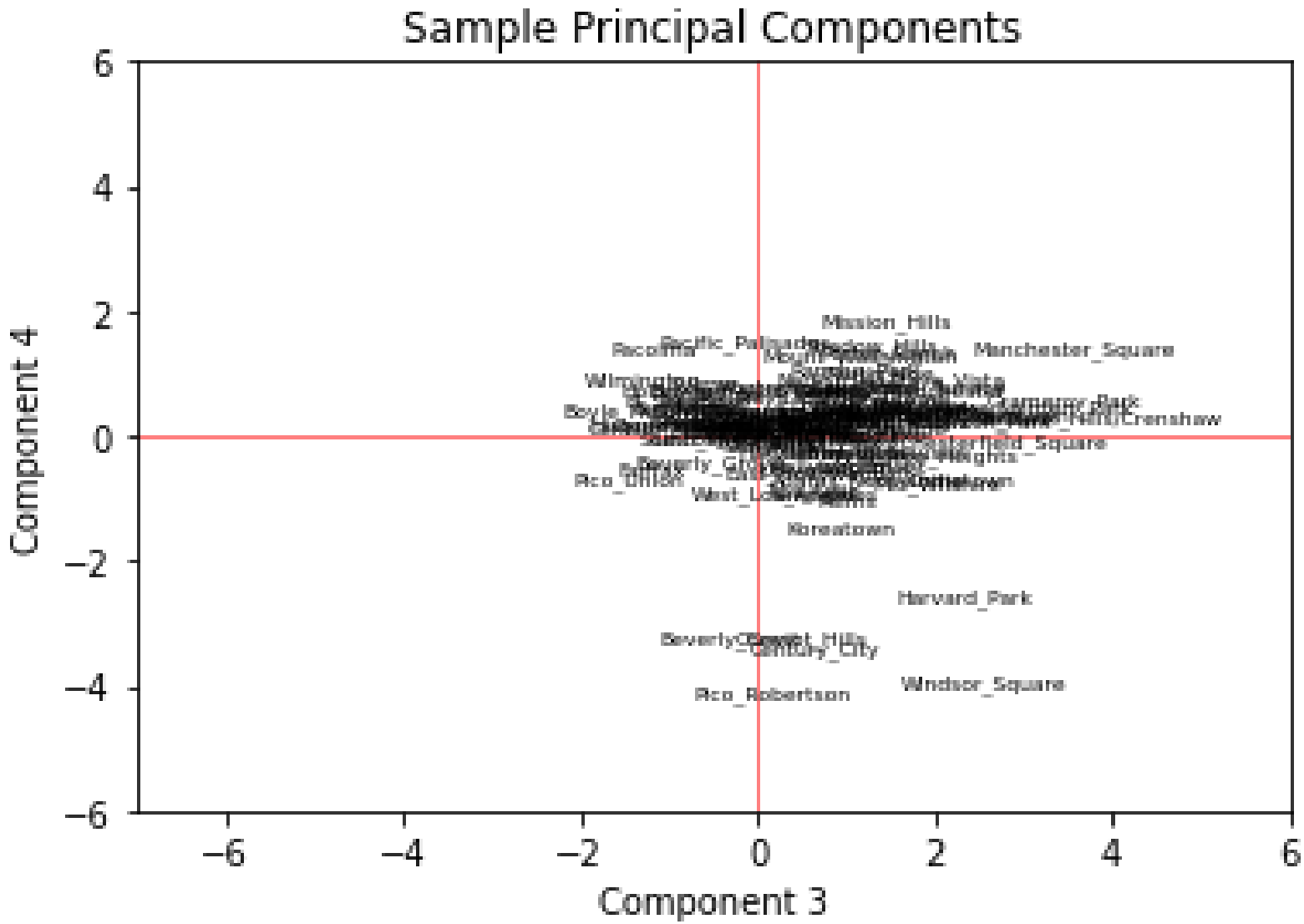


Figure 3.5.2 PC Scores Plot

Principal Component Regression

- In regression analysis, when there is strong multicollinearity among independent variables, the results from classical regression are generally poor.
- In this case, we can use the first m principal components $z_1, \cdots, z_m$ of the original variables $x_1, \cdots, x_p$ (whose cumulative contribution rate reaches a certain level) to establish a **principal component regression** model:

$$y = \beta_0 + \beta_1 z_1 + \cdots + \beta_m z_m$$

- Compute the scores of the first m principal components from the observed data matrix of the original variables, use them as observed values for $z_1, \cdots, z_m$, and establish the regression model of y on $z_1, \cdots, z_m$ to obtain the PCR equation.

- This simplifies the regression structure and eliminates the influence of variable correlation.
- Since PCs are linear combinations, not directly observed, their meaning is sometimes unclear.
- After obtaining the PCR equation, it is generally necessary to transform it back to a regression equation in terms of the original variables via inverse transformation.

3.5 Applications of PCA

Example 5 (PCR for French Economic Data) Consider the relationship between total imports (y) and three independent variables: GDP (x1), inventory (x2), and total consumption (x3) (all in billion francs). Data collected for 11 years from 1949 to 1959 are shown in Table 3.7. Perform PCR on this data.

Table 3.7 French economic analysis data.

Serial No.	x_1	x_2	x_3	y
1	149.3	4.2	108.1	15.9
2	161.2	4.1	114.8	16.4
3	171.5	3.1	123.2	19.0
4	175.5	3.1	126.9	19.1
5	180.8	1.1	132.1	18.8
6	190.7	2.2	137.7	20.4
7	201.1	2.1	146.0	22.7
8	212.4	5.6	154.1	26.5
9	226.1	5.0	162.3	28.1
10	231.9	5.1	164.3	27.6
11	239.0	0.7	167.6	26.3

Python Code for Example 5

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression

# Create data
df = pd.DataFrame({
    'x1': [149.3, 161.2, 171.5, 175.5, ...],
    'x2': [4.2, 4.1, 3.1, 3.1, ...],
    'x3': [108.1, 114.8, 123.2, 126.9, ...],
    'y': [15.9, 16.4, 19.0, 19.1, ...]
})

# OLS
X = df[['x1', 'x2', 'x3']]
y = df['y']
ols = LinearRegression().fit(X, y)
print("OLS Summary:")
print("Coefs:", ols.coef_)
print("Intercept:", ols.intercept_)

# PCR
scaler = StandardScaler()
X_sc = scaler.fit_transform(X)
pca = PCA()
pca.fit(X_sc)

# PCA results
print("\nPCA Summary:")
print("Var Ratio:", pca.explained_variance_ratio_)
print("Loadings:", pca.components_)

# Predict PCs
pre = pca.transform(X_sc)
df['z1'] = pre[:, 0]  # 1st PC
df['z2'] = pre[:, 1]  # 2nd PC

# Regress y on PCs
pc_reg = LinearRegression().fit(df[['z1', 'z2']], y)
print("\nPCR Summary:")
print("Coefs:", pc_reg.coef_)
print("Intercept:", pc_reg.intercept_)

# Transform back
beta = pc_reg.coef_
A = pca.components_
x_m = scaler.mean_
x_s = scaler.scale_
coef = (beta[0] * A[0, :] + beta[1] * A[1, :]) / x_s
beta0 = pc_reg.intercept_ - np.sum(x_m * coef)
print("\nTransformed Coefs:")
print(np.concatenate(([beta0], coef)))
```

For comparison, we first use the general linear regression analysis method. From the calculation results, the regression equation with three variables is

$$y = -10.12799 - 0.0514x_1 + 0.58695x_2 + 0.28685x_3$$

- Carefully observing the above equation, we find it is unreasonable.
- Because y is the import volume, and x_1 is the gross domestic product (GDP), but its coefficient is negative, meaning that the higher the GDP, the less the import volume, which is inconsistent with reality.
- Through the principal component analysis of the three explanatory variables x_1, x_2, x_3, we find that there is multicollinearity among them (because the smallest eigenvalue of their sample correlation matrix is close to 0).
- To overcome the influence of multicollinearity, we use the principal component regression method below.

- Since the cumulative contribution rate of the first two principal components has reached 99.9%, we take the number of principal components as 2.
- To perform principal component regression, we first calculate the score matrix of the first two principal components corresponding to the observation data matrix of the three variables, and then perform regression analysis on the principal components combined with the observed values of y.
- From the calculation results, we obtain the following regression equation

$$y = 21.8909 + 2.9892z_1^* - 0.8288z_2^*$$

- The coefficients of this regression equation have passed the test and are significant. However, this equation gives the relationship between the response variable y and the principal components z_1, z_2. We wish to establish the relationship between y and the original variables x_1, x_2 and x_3.

Since

$$y = \beta_0^* + \beta_1^* z_1^* + \beta_2^* z_2^*$$

$$z_i^* = u_{i1} x_1^* + u_{i2} x_2^* + u_{i3} x_3^*$$

$$= \frac{u_{i1}(x_1 - \overline{x}_1)}{\sqrt{s_{11}}} + \frac{u_{i2}(x_2 - \overline{x}_2)}{\sqrt{s_{22}}} + \frac{u_{i3}(x_3 - \overline{x}_3)}{\sqrt{s_{33}}}, i = 1, 2,$$

Substituting the corresponding data, the obtained regression equation is

$$y = -9.13010782 + 0.07277981\, x_1 + 0.60922012\, x_2 + 0.10625939\, x_3,$$

where all regression coefficients are positive, which is more reasonable than the regression equation obtained by the general linear regression method.

Hierarchical Clustering based on PCA

This introduces hierarchical clustering based on PCA, belonging to the category of non-classical multivariate data analysis methods. The statistical ideas are the same as classical methods, but results are improved through computational procedures, emphasizing visualization and geometric intuition, especially 2D graphics. Whether data is high-dimensional or hierarchical, relationships are displayed intuitively graphically. Graphics leave a deeper impression than numbers.

Using PCA-based hierarchical clustering is because direct clustering results are often poor due to noise in high-dimensional data with complex relationships. If we first reduce dimensionality via PCA and then perform cluster analysis on the reduced data, results may improve.

3.5 Applications of PCA

Example 6 Cluster analysis of US production data. Economic data for 48 US states over 17 years (1970-1986). Variables: state, year, private capital (pcap), roads (hwy), water/sewer systems (water), other utilities (util), public capital (pc), gross state product (gsp), non-agricultural employment (emp), unemployment rate (unemp). Use the 17-year average for each state.

- We first compute the annual averages to create a new dataset (z), then perform Principal Component Analysis, followed by cluster analysis based on the PCA results.
- The results are shown in Figures 3.5, 3.6, and 3.7. Figure 3.5 is the dendrogram for selecting the number of clusters, with a small bar plot of eigenvalues in the upper right corner.
- Figure 3.6 is the 3D dendrogram on the plane of the first two principal components. Figure 3.7 shows the three clusters automatically identified, marked on the principal component plane.

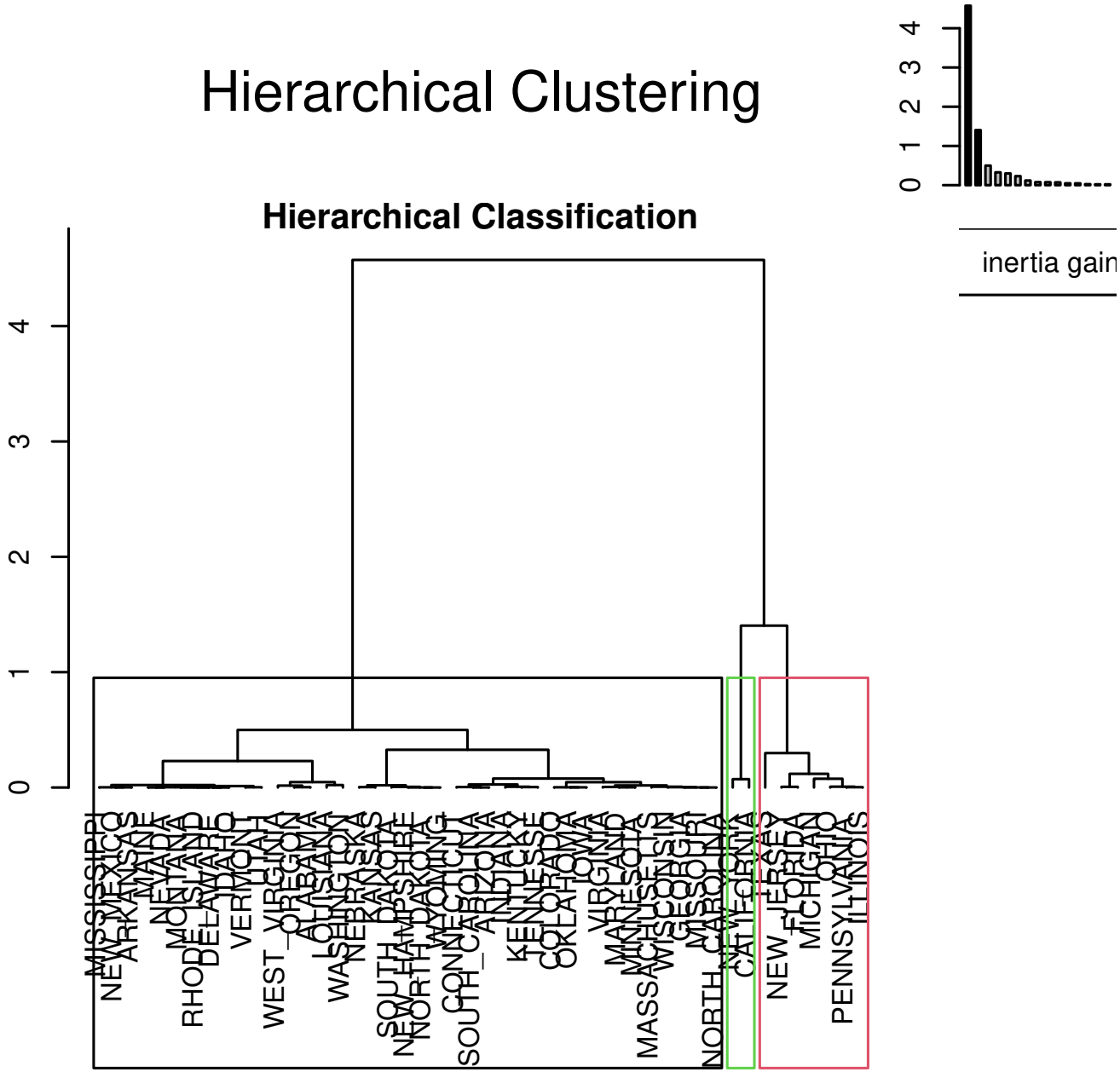


Figure 3.5. The dendrogram for selecting the number of clusters.

Hierarchical clustering on the factor map

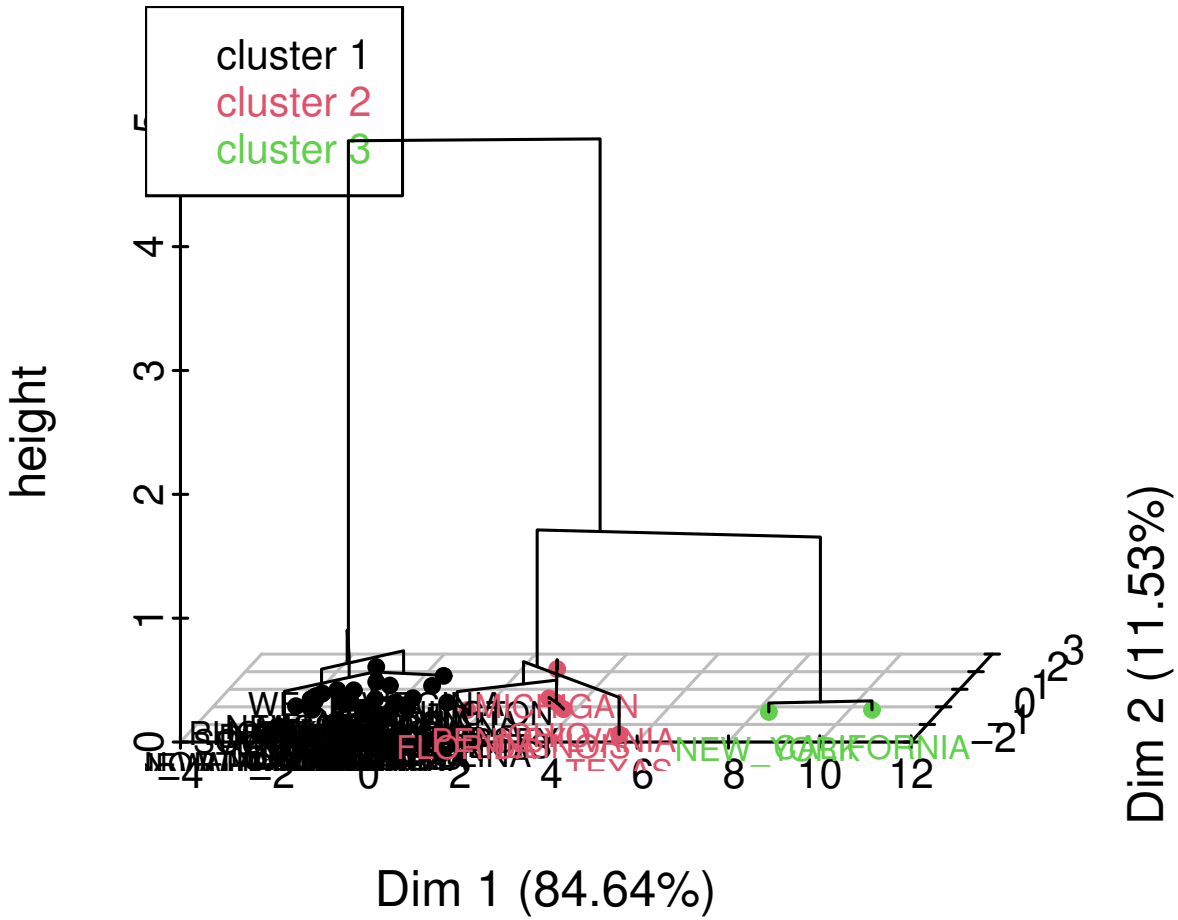


Figure 3.6. The 3D dendrogram on the plane of the first two principal

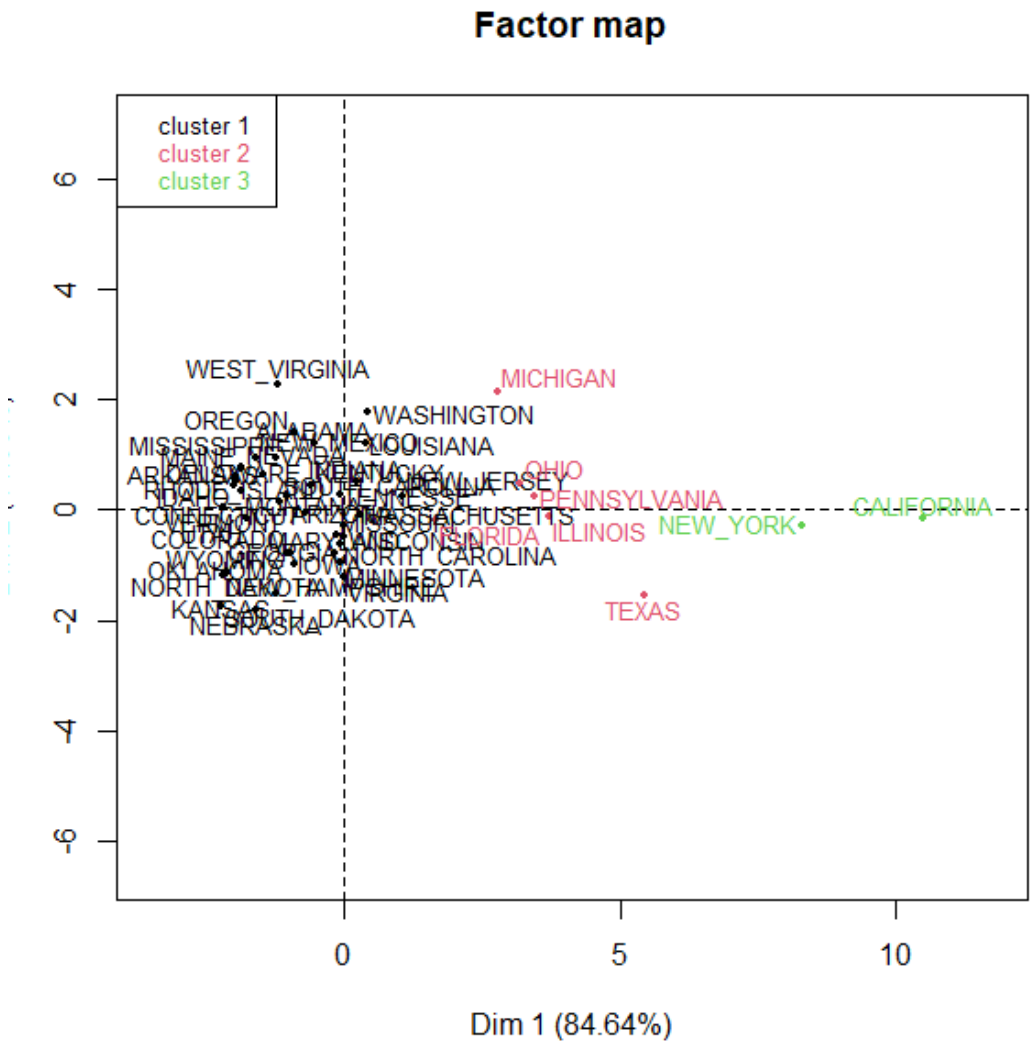


Figure 3.7. The three clusters automatically identified, marked on the principal component plane.