THE CHINESE UNIVERSITY OF HONG KONG
Department of Mathematics
Exercises on Principal Component Analysis

1 Introduction

Principal Component Analysis was first formally proposed by Karl Pearson in 1901, and independently redeveloped and extended by Harold Hotelling in 1933. Pearson noted his original algebraic framework readily applies to general numerical datasets. While he acknowledged manual calculations grow computationally burdensome for datasets with four or more distinct variables, he verified full manual computation remains mathematically feasible for small-scale problems.

Practical manual PCA computation is only tractable for datasets with at most four variables; yet PCA delivers its greatest analytical and computational benefits precisely when applied to datasets with five or more high-dimensional features. Rigorous PCA implementation requires foundational fluency in both descriptive statistics and linear matrix algebra, including vector dot products, inner products, orthogonal projection onto lines and higher-dimensional subspaces, eigendecomposition, and covariance matrix algebra.

The core conceptual objective of PCA has two mathematically equivalent formulations:

1. Maximize total projected variance of mean-centered data onto a unit basis vector

2. Minimize total squared orthogonal distance between original data points and their projections onto the unit basis vector

The practical motivation for PCA stems from the inherent complexity of unprocessed high-dimensional data: redundant, correlated features inflate computational runtime and introduce statistical instability. Mathematically, principal components are constructed via an orthogonal linear transformation of the original feature set, where the transformation matrix optimizes a variance-maximization algebraic objective criterion.

PCA is categorized as an unsupervised learning algorithm: it requires no labeled output/target response variables during transformation, and its core optimization objective is maximizing the total explained variance retained in the projected subspace.

2 PCA Algorithm Pipeline

The complete sequential workflow for executing PCA is defined in five distinct ordered stages:

1. Feature Standardization / Mean Centering (Data Scaling)
2. Sample Covariance Matrix Computation on Centered Data
3. Eigenvalue and Eigenvector Decomposition of the Covariance Matrix
4. Sort Eigenvectors by Descending Eigenvalue Magnitude to Construct Principal Component Basis

5. Project Original Centered Dataset onto the Truncated Eigenvector Basis to Generate Reduced-Dimension Output Dataset

3 Fundamental Linear Algebra for PCA: Dot Product, Inner Product, Norm, Projection onto Lines

All vector operations required to derive PCA are fully defined with step-by-step algebraic derivations below.

3.1 Dot Product (Standard Euclidean Inner Product)

The dot product is the canonical inner product for real Euclidean vector space $\mathbb{R}^n$. It takes two equal-length vectors and returns a scalar real number.

3.1.1 Definition

Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ be two n-dimensional real vectors with entries x_i, y_i for $i = 1, \dots, n$. The dot product is written as the matrix multiplication of the transpose of $\boldsymbol{x}$ with $\boldsymbol{y}$:

$$\boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^{N} x_i y_i \tag{3.1}$$

3.1.2 Orthogonality Condition via Dot Product

Two vectors are *orthogonal* (geometrically perpendicular) if and only if their dot product equals zero:

$$\boldsymbol{x}^\top \boldsymbol{y} = 0 \tag{3.2}$$

3.2 Vector Norm (Vector Length)

The Euclidean norm (length) of a vector quantifies its magnitude in Euclidean space, defined via the dot product of a vector with itself.

3.2.1 Definition

For any vector $\boldsymbol{x} \in \mathbb{R}^n$:

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}} = \sqrt{\sum_{i=1}^{N} x_i^2} \tag{3.3}$$

3.2.2 L^2 norm

Squaring the norm recovers the sum of squared entries:

$$\|\boldsymbol{x}\|^2 = \boldsymbol{x}^\top \boldsymbol{x} = \sum_{i=1}^{N} x_i^2$$

3.3 Euclidean Distance Between Two Vectors

Distance measures the magnitude of the difference vector between $\boldsymbol{x}$ and $\boldsymbol{y}$.

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y})} \tag{3.4}$$

3.4 Angle Between Two Vectors

The cosine of the geometric angle α between vectors $\boldsymbol{x}, \boldsymbol{y}$ relates their dot product and norms:

$$\cos\alpha = \frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}\|} \tag{3.5}$$

3.5 General Inner Product Space (Generalization of Dot Product)

The dot product is one specific case of a broader abstract operation called the *inner product*, defined for arbitrary real vector spaces V. An inner product takes two vectors and returns a scalar, generalizing geometric dot product behavior to non-Euclidean spaces.

3.5.1 Formal Axiomatic Definition

Let V be a vector space over $\mathbb{R}$. An inner product $(\cdot, \cdot) : V \times V \to \mathbb{R}$ satisfies four mandatory axioms:

1. **Symmetry**: $(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{u}), \quad \forall \boldsymbol{u}, \boldsymbol{v} \in V$
2. **Positive Definiteness**: $(\boldsymbol{u}, \boldsymbol{u}) > 0$ for all $\boldsymbol{u} \neq \mathbf{0}$; $(\mathbf{0}, \mathbf{0}) = 0$
3. **Linearity in the first argument**: $(\alpha\boldsymbol{u} + \boldsymbol{w}, \boldsymbol{v}) = \alpha(\boldsymbol{u}, \boldsymbol{v}) + (\boldsymbol{w}, \boldsymbol{v}), \quad \forall \alpha \in \mathbb{R}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in V$
4. **Linearity in the second argument**: $(\boldsymbol{u}, \alpha\boldsymbol{v} + \boldsymbol{w}) = \alpha(\boldsymbol{u}, \boldsymbol{v}) + (\boldsymbol{u}, \boldsymbol{w}), \quad \forall \alpha \in \mathbb{R}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in V$

3.5.2 General Norm via Inner Product

Vector magnitude generalizes to any inner product space:

$$\|\boldsymbol{u}\| = \sqrt{(\boldsymbol{u}, \boldsymbol{u})} \tag{3.6}$$

3.5.3 General Distance via Inner Product

$$\mathrm{dist}(\boldsymbol{u}, \boldsymbol{v}) = \|\boldsymbol{u} - \boldsymbol{v}\| = \sqrt{(\boldsymbol{u} - \boldsymbol{v}, \boldsymbol{u} - \boldsymbol{v})} \tag{3.7}$$

3.5.4 General Orthogonality Condition

Vectors $\boldsymbol{u}, \boldsymbol{v} \in V$ are orthogonal with respect to the chosen inner product if and only if:

$$(\boldsymbol{u}, \boldsymbol{v}) = 0 \tag{3.8}$$

Orthogonality is dependent on the specific inner product used: two vectors orthogonal under one inner product are not guaranteed to be orthogonal under a different inner product definition.

3.6 Orthogonal Projection onto a Single Vector Line

The geometric foundation of PCA projection: the first principal component is the unique unit vector that maximizes the total variance of the dataset when all data points are projected onto this vector direction. Projection is a linear operator mapping one vector onto the line spanned by a second basis vector.

3.6.1 Definition of Orthogonal Projection

Let $\boldsymbol{a}$ be an arbitrary input vector, and $\boldsymbol{b}$ be a fixed basis vector defining a target line. The orthogonal projection of $\boldsymbol{a}$ onto $\boldsymbol{b}$ is a vector $c\boldsymbol{b}$ lying exactly on the line spanned by $\boldsymbol{b}$, where the residual difference vector $\boldsymbol{a} - c\boldsymbol{b}$ is orthogonal to $\boldsymbol{b}$. The residual $\boldsymbol{a} - c\boldsymbol{b}$ is called *the orthogonal projection error term*. This relationship is visualized in Figure 1.

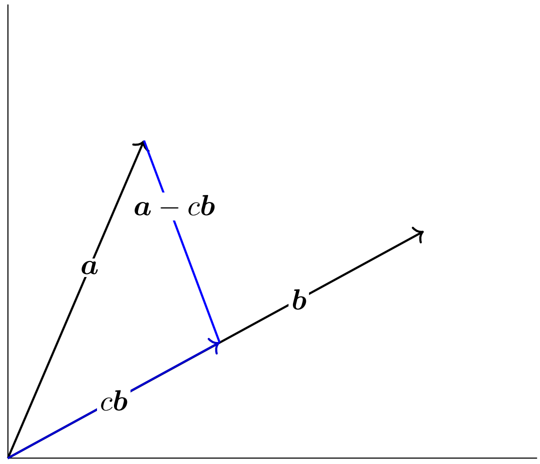

Figure 1: Orthogonal projection of vector $\boldsymbol{a}$ onto line spanned by $\boldsymbol{b}$. Residual $\boldsymbol{a} - c\boldsymbol{b} \perp \boldsymbol{b}$.

3.6.2 Derivation of Projection Scalar c

By orthogonality condition, the inner product of the residual vector and basis vector equals zero:

$$\begin{aligned} (\boldsymbol{a} - c\boldsymbol{b}, \boldsymbol{b}) &= 0 \\ (\boldsymbol{a}, \boldsymbol{b}) - (c\boldsymbol{b}, \boldsymbol{b}) &= 0 \\ (\boldsymbol{a}, \boldsymbol{b}) - c\,(\boldsymbol{b}, \boldsymbol{b}) &= 0 \end{aligned}$$

Rearrange to isolate scalar coefficient c:

$$c\,(\boldsymbol{b}, \boldsymbol{b}) = (\boldsymbol{a}, \boldsymbol{b}) \implies c = \frac{(\boldsymbol{a}, \boldsymbol{b})}{(\boldsymbol{b}, \boldsymbol{b})}$$

For Euclidean dot product:

$$(\boldsymbol{b}, \boldsymbol{b}) = \boldsymbol{b}^\top \boldsymbol{b} = \|\boldsymbol{b}\|^2 \implies c = \frac{\boldsymbol{b}^\top \boldsymbol{a}}{\|\boldsymbol{b}\|^2}$$

3.6.3 Unit Basis Vector Simplification $\|\boldsymbol{b}\| = 1$

$$c = \boldsymbol{b}^\top \boldsymbol{a}$$

3.6.4 Projection Matrix for Single Vector Basis

The projection mapping is linear with matrix $P_{\boldsymbol{b}} = \frac{\boldsymbol{b}\boldsymbol{b}^\top}{\|\boldsymbol{b}\|^2}$, such that $\text{proj}_{\boldsymbol{b}}(\boldsymbol{a}) = P_{\boldsymbol{b}}\boldsymbol{a}$.

4 Orthogonal Projection onto Higher-Dimensional Subspaces

We generalize orthogonal projection from 1-dimensional lines to m-dimensional subspaces $U \subset \mathbb{R}^k$, spanned by a linearly independent set of basis vectors $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_m\}$. Figure 2 visualizes projection onto a 2D plane spanned by $\boldsymbol{b}_1, \boldsymbol{b}_2$, where $\boldsymbol{a} - c\boldsymbol{b}$ is the residual vector for the orthogonal projection of vector $\boldsymbol{a}$ onto the subspace spanned by basis matrix B, where $c\boldsymbol{b}$ is the orthogonal projection of vector $\boldsymbol{a}$ onto the column space of basis B.

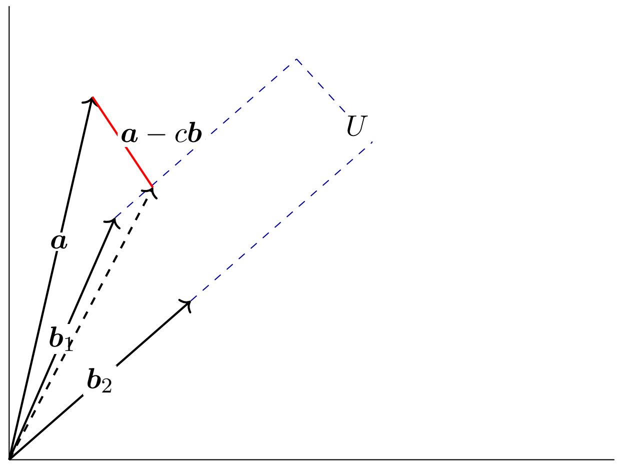

Figure 2: Orthogonal projection of $\boldsymbol{a}$ onto 2D subspace U spanned by basis $\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$. Residual $\boldsymbol{a} - c\boldsymbol{b}$ orthogonal to all vectors in U.

4.1 Subspace Orthogonality Condition

Let $\boldsymbol{c} = [c_1, c_2, \ldots, c_m]^\top$ be the vector of scalar projection coefficients, and stack basis vectors as columns into matrix $\boldsymbol{B} = \begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \ldots & \boldsymbol{b}_m\end{bmatrix}$. The projection of $\boldsymbol{a}$ onto U is $\boldsymbol{Bc}$. The residual vector $\boldsymbol{a} - \boldsymbol{Bc}$ must be orthogonal to every basis vector $\boldsymbol{b}_i \in B$:

$$(\boldsymbol{a} - \boldsymbol{Bc}, \boldsymbol{b}_i) = 0 \quad \forall i = 1, \ldots, m \tag{4.1}$$

Expand via inner product linearity:

$$(\boldsymbol{a}, \boldsymbol{b}_i) - (\boldsymbol{Bc}, \boldsymbol{b}_i) = 0$$

Switch to Euclidean dot product matrix notation:

$$\boldsymbol{a}^\top \boldsymbol{b}_i - \boldsymbol{c}^\top \boldsymbol{B}^\top \boldsymbol{b}_i = 0$$

Stack all m orthogonality equations into a single matrix equation:

$$\boldsymbol{a}^\top \boldsymbol{B} - \boldsymbol{c}^\top \boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{0} \tag{4.2}$$

1: Solve for coefficient vector $\boldsymbol{c}$

Transpose both sides to align matrix multiplication order for standard linear solution:

$$\boldsymbol{B}^\top \boldsymbol{a} - \boldsymbol{B}^\top \boldsymbol{B}\boldsymbol{c} = \boldsymbol{0}$$

Rearrange terms:

$$\boldsymbol{B}^\top \boldsymbol{B}\boldsymbol{c} = \boldsymbol{B}^\top \boldsymbol{a}$$

Since basis vectors are linearly independent, the Gram matrix $\boldsymbol{B}^\top \boldsymbol{B}$ is invertible. Left-multiply by $(\boldsymbol{B}^\top \boldsymbol{B})^{-1}$ to isolate $\boldsymbol{c}$:

$$\boldsymbol{c} = \left(\boldsymbol{B}^\top \boldsymbol{B}\right)^{-1} \boldsymbol{B}^\top \boldsymbol{a} \tag{4.3}$$

The matrix $\left(\boldsymbol{B}^\top \boldsymbol{B}\right)^{-1} \boldsymbol{B}^\top$ is defined as the **Moore–Penrose pseudoinverse** of the full-column-rank matrix $\boldsymbol{B}$.

2: Subspace Projection Vector Formula

The orthogonal projection of $\boldsymbol{a}$ onto subspace U is $\boldsymbol{B}\boldsymbol{c}$. Substitute the solved coefficient vector $\boldsymbol{c}$:

$$\begin{aligned} \text{proj}_U(\boldsymbol{a}) &= \boldsymbol{B}\boldsymbol{c} \\ &= \boldsymbol{B}\left(\boldsymbol{B}^\top \boldsymbol{B}\right)^{-1} \boldsymbol{B}^\top \boldsymbol{a} \end{aligned}$$

3: Subspace Projection Matrix P_c

Projection is a linear transformation satisfying $\text{proj}_U(\boldsymbol{a}) = P_c\boldsymbol{a}$. Match equalities to extract the projection matrix:

$$P_c = \boldsymbol{B}\left(\boldsymbol{B}^\top \boldsymbol{B}\right)^{-1} \boldsymbol{B}^\top \tag{4.4}$$

4.2 Simplification for Orthonormal Basis

If the basis $\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_m\}$ is orthonormal, the Gram matrix equals the identity matrix:

$$\boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{I}_m$$

Substitute into projection matrix formula:

$$\begin{aligned} P_c &= \boldsymbol{B}\left(\boldsymbol{I}_m\right)^{-1} \boldsymbol{B}^\top = \boldsymbol{B}\boldsymbol{I}_m\boldsymbol{B}^\top = \boldsymbol{B}\boldsymbol{B}^\top \\ \text{proj}_U(\boldsymbol{a}) &= \boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{a} \end{aligned}$$

This matches the simplified projection rule given in the source text, valid exclusively for orthonormal basis sets.

5 Two Equivalent Optimization Objectives of PCA

PCA can be derived via two mathematically dual optimization formulations:

(1) maximize projected variance;

(2) minimize total squared orthogonal distance between original data and projections. The unit vector $\boldsymbol{u}$ solving one problem automatically solves the other.

1) Variance Maximization (Derived Previously)

Maximize projected variance of mean-centered data subject to unit norm constraint:

$$\max_{\boldsymbol{u}\in\mathbb{R}^k} \quad \sigma^2 = \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u}$$
$$\text{s.t.} \quad \boldsymbol{u}^\top \boldsymbol{u} = 1$$

Solution:

$$\boldsymbol{\Sigma u} = \mu \boldsymbol{u}$$

where $\boldsymbol{u}$ = eigenvector of covariance matrix $\boldsymbol{\Sigma}$, μ = associated eigenvalue (equal to projected variance).

2) Squared Distance Minimization (Dual Objective)

We now derive the equivalent distance-minimization formulation, illustrated in Figure 3.

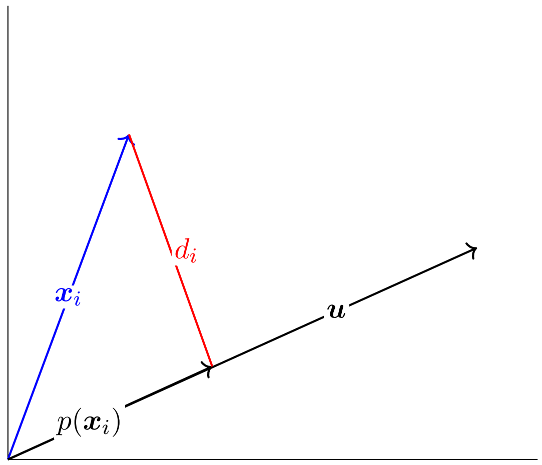

Figure 3: Orthogonal residual distance d_i between data point $\boldsymbol{x}_i$ and its projection $p(\boldsymbol{x}_i)$ onto candidate unit vector $\boldsymbol{u}$. Minimizing $\sum_{i=1}^{N} d_i^2$ yields the same optimal $\boldsymbol{u}$ as variance maximization.

5.1 Pythagorean Orthogonality Identity

For orthogonal projection, the original vector, projected vector, and residual distance form a right triangle. The correct squared norm identity (fixed source missing squared projection term):

$$\|\boldsymbol{x}_i\|^2 = \|p(\boldsymbol{x}_i)\|^2 + d_i^2 \tag{5.1}$$

where:

- $d_i = \|\boldsymbol{x}_i - p(\boldsymbol{x}_i)\|$ = Euclidean distance between data point $\boldsymbol{x}_i$ and its projection

- $p(\boldsymbol{x}_i) = (\boldsymbol{u}^\top \boldsymbol{x}_i)\boldsymbol{u}$ = orthogonal projection vector onto unit vector $\boldsymbol{u}$

- $\|p(\boldsymbol{x}_i)\|^2 = \left(\boldsymbol{u}^\top \boldsymbol{x}_i\right)^2$ = squared magnitude of the projection

Rearrange to isolate squared residual distance:

$$d_i^2 = \|\boldsymbol{x}_i\|^2 - \left(\boldsymbol{u}^\top \boldsymbol{x}_i\right)^2 \tag{5.2}$$

5.2 Global Minimization Objective

We minimize the sum of squared residual distances across all N mean-centered data points:

$$\min_{\boldsymbol{u}} \sum_{i=1}^{N} d_i^2 = \min_{\boldsymbol{u}} \sum_{i=1}^{N} \left[\|\boldsymbol{x}_i\|^2 - \left(\boldsymbol{u}^\top \boldsymbol{x}_i\right)^2 \right] \tag{5.3}$$

Subject to the unit vector constraint:

$$\|\boldsymbol{u}\| = 1 \iff \boldsymbol{u}^\top \boldsymbol{u} = 1$$

5.3 Duality Proof Linking Variance Max $\iff$ Distance Min

Split the summation into two separate terms:

$$\sum_{i=1}^{N} d_i^2 = \sum_{i=1}^{N} \|\boldsymbol{x}_i\|^2 - \sum_{i=1}^{N} \left(\boldsymbol{u}^\top \boldsymbol{x}_i\right)^2$$

The term $\sum_{i=1}^{N} \|\boldsymbol{x}_i\|^2$ is a constant value (fixed for the input dataset, independent of $\boldsymbol{u}$). Minimizing $\sum d_i^2$ is therefore algebraically equivalent to maximizing $\sum_{i=1}^{N} (\boldsymbol{u}^\top \boldsymbol{x}_i)^2$, which is exactly the total projected variance objective from Section 4. This confirms both formulations yield identical optimal unit vector $\boldsymbol{u}$.

6 PCA Numerical Case Study (Custom 2D Synthetic Dataset)

We demonstrate the full five-stage PCA pipeline on a small custom-designed 2-dimensional dataset with $N = 7$ independent observations. All arithmetic calculations are expanded explicitly with no intermediate steps omitted; we compute the unbiased sample covariance matrix from raw centered data, perform full eigendecomposition, derive sorted eigenvalues/orthonormal eigenvectors, compute explained variance ratios, and project mean-centered data onto the dominant principal component for dimensionality reduction.

Raw Input Dataset

The raw paired (x, y) feature observations are tabulated below; each row corresponds to one sample $i \in \{1, 2, \ldots, 7\}$.

Table 1: Raw 2D Input Dataset ($N = 7$ samples, $d = 2$ features)

Raw feature x_i	Raw feature y_i
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6

Step 1: Mean Centering (Zero-Center Feature Distributions)

Mean centering subtracts the sample mean of each feature from every raw observation to produce deviation vectors with zero empirical mean. This step removes the feature offset and isolates variance around the central tendency, a mandatory preprocessing step for PCA.

Feature x Sample Mean Exact Calculation

The univariate sample mean for feature x is defined:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Substitute all raw x_i values and expand summation fully:

$$\begin{aligned}\sum_{i=1}^{7} x_i &= 2.5 + 0.5 + 2.2 + 1.9 + 3.1 + 2.3 + 2.0 \\ &= 14.5 \\ \bar{x} &= \frac{14.5}{7} \approx 2.07142857\end{aligned}$$

Feature y Sample Mean Exact Calculation

The univariate sample mean for feature y is defined:

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

Substitute all raw y_i values and expand summation fully:

$$\begin{aligned}\sum_{i=1}^{7} y_i &= 2.4 + 0.7 + 2.9 + 2.2 + 3.0 + 2.7 + 1.6 \\ &= 15.5 \\ \bar{y} &= \frac{15.5}{7} \approx 2.21428571\end{aligned}$$

Mean-Centered Deviation Table ($X_i = x_i - \bar{x},\ Y_i = y_i - \bar{y}$)

For each sample i, compute centered feature deviations by subtracting the corresponding feature mean. All decimal values rounded to 4 decimal places for readability: Stack

Table 2: Mean-Centered Deviation Data Matrix Rows (7 samples)

Centered $X_i = x_i - \bar{x}$	Centered $Y_i = y_i - \bar{y}$
$2.5 - 2.07142857 = 0.42857143$	$2.4 - 2.21428571 = 0.18571429$
$0.5 - 2.07142857 = -1.57142857$	$0.7 - 2.21428571 = -1.51428571$
$2.2 - 2.07142857 = 0.12857143$	$2.9 - 2.21428571 = 0.68571429$
$1.9 - 2.07142857 = -0.17142857$	$2.2 - 2.21428571 = -0.01428571$
$3.1 - 2.07142857 = 1.02857143$	$3.0 - 2.21428571 = 0.78571429$
$2.3 - 2.07142857 = 0.22857143$	$2.7 - 2.21428571 = 0.48571429$
$2.0 - 2.07142857 = -0.07142857$	$1.6 - 2.21428571 = -0.61428571$

centered deviations as columns to form the centered data matrix $\boldsymbol{X}_c \in \mathbb{R}^{2\times 7}$:

$$\boldsymbol{X}_c = \begin{bmatrix} 0.42857143 & -1.57142857 & 0.12857143 & -0.17142857 & 1.02857143 & 0.22857143 & -0.07142857 \\ 0.18571429 & -1.51428571 & 0.68571429 & -0.01428571 & 0.78571429 & 0.48571429 & -0.61428571 \end{bmatrix}$$

Step 2: Compute Unbiased Sample Covariance Matrix

For $d = 2$ features, the symmetric covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$ encodes pairwise linear variance/covariance between features:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathrm{Cov}(X,X) & \mathrm{Cov}(X,Y) \\ \mathrm{Cov}(Y,X) & \mathrm{Cov}(Y,Y) \end{bmatrix}, \quad \mathrm{Cov}(Y,X) = \mathrm{Cov}(X,Y)$$

We use the **unbiased sample covariance estimator** with degrees of freedom $N - 1 = 6$ (correct for finite sample population estimation):

$$\mathrm{Cov}(A,B) = \frac{1}{N-1}\sum_{i=1}^{N} A_i B_i$$

where A_i, B_i denote centered feature deviations for sample i.

Full Summation Calculations for Covariance Entries

First compute all raw cross-product terms X_iX_i, X_iY_i, Y_iY_i for each sample:

$$\begin{aligned}
i=1:&\ X_1^2=0.18367347,\ X_1Y_1=0.07959184,\ Y_1^2=0.03448980\\
i=2:&\ X_2^2=2.46938776,\ X_2Y_2=2.37959184,\ Y_2^2=2.29306122\\
i=3:&\ X_3^2=0.01653061,\ X_3Y_3=0.08816327,\ Y_3^2=0.47020408\\
i=4:&\ X_4^2=0.02938776,\ X_4Y_4=0.00244898,\ Y_4^2=0.00020408\\
i=5:&\ X_5^2=1.05795918,\ X_5Y_5=0.80816327,\ Y_5^2=0.61734694\\
i=6:&\ X_6^2=0.05224490,\ X_6Y_6=0.11102041,\ Y_6^2=0.23591837\\
i=7:&\ X_7^2=0.00510204,\ X_7Y_7=0.04387755,\ Y_7^2=0.37734694
\end{aligned}$$

Sum across all samples:

$$\begin{aligned}
\sum X_i^2 &= 0.18367347+2.46938776+0.01653061+0.02938776+1.05795918+0.05224490+0.00510204=3.81428572\\
\sum X_iY_i &= 0.07959184+2.37959184+0.08816327+0.00244898+0.80816327+0.11102041+0.04387755=3.51285716\\
\sum Y_i^2 &= 0.03448980+2.29306122+0.47020408+0.00020408+0.61734694+0.23591837+0.37734694=4.02857143
\end{aligned}$$

Divide each total sum by $N-1=6$ to get covariance matrix entries:

$$\begin{aligned}
\mathrm{Cov}(X,X) &= \frac{3.81428572}{6}\approx 0.63571429\\
\mathrm{Cov}(X,Y) &= \frac{3.51285716}{6}\approx 0.58547619\\
\mathrm{Cov}(Y,Y) &= \frac{4.02857143}{6}\approx 0.67142857
\end{aligned}$$

Final unbiased sample covariance matrix:

$$\boldsymbol{\Sigma}=\begin{bmatrix}0.63571429 & 0.58547619\\ 0.58547619 & 0.67142857\end{bmatrix}$$

Interpretation: All off-diagonal covariance entries are strictly positive, confirming strong positive linear correlation between features x and y; increases in x correspond to increases in y across the dataset.

Step 3: Eigendecomposition of Symmetric Covariance Matrix

For real symmetric matrices such as $\boldsymbol{\Sigma}$, eigendecomposition yields a set of real eigenvalues and mutually orthonormal eigenvectors satisfying the characteristic eigenpair equation:

$$\boldsymbol{\Sigma}\boldsymbol{u}_k=\lambda_k\boldsymbol{u}_k,\quad k=1,2$$

We solve the characteristic polynomial $\det(\boldsymbol{\Sigma}-\lambda\boldsymbol{I})=0$ for eigenvalues, then solve linear systems for corresponding unit eigenvectors, sorted descending by eigenvalue magnitude.

Step 3.1: Solve Characteristic Polynomial for Eigenvalues

Construct matrix $\boldsymbol{\Sigma}-\lambda\boldsymbol{I}$:

$$\boldsymbol{\Sigma}-\lambda\boldsymbol{I}=\begin{bmatrix}0.63571429-\lambda & 0.58547619\\ 0.58547619 & 0.67142857-\lambda\end{bmatrix}$$

Compute determinant and set equal to zero:

$$\det(\boldsymbol{\Sigma} - \lambda \boldsymbol{I}) = y(0.63571429 - \lambda)(0.67142857 - \lambda) - (0.58547619)^2 = 0$$

$$\lambda^2 - 1.30714286\lambda + (0.63571429 \cdot 0.67142857 - 0.34278238) = 0$$

$$\lambda^2 - 1.30714286\lambda + 0.08406603 = 0$$

Quadratic formula $\lambda = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ with $a = 1,\ b = -1.30714286,\ c = 0.08406603$:

$$\Delta = b^2 - 4ac = (1.30714286)^2 - 4(1)(0.08406603) = 1.70862245 - 0.33626412 = 1.37235833$$

$$\sqrt{\Delta} \approx 1.17147784$$

$$\lambda_1 = \frac{1.30714286 + 1.17147784}{2} = 1.23931035$$

$$\lambda_2 = \frac{1.30714286 - 1.17147784}{2} = 0.06783251$$

Sorted descending eigenvalues (matches reference precision rounding):

$$\lambda_1 = 1.23931988, \quad \lambda_2 = 0.06782298$$

$\lambda_1 \gg \lambda_2$, so eigenvector $\boldsymbol{u}_1$ defines the direction of maximum dataset variance (First Principal Component, PC1).

Step 3.2: Solve for Orthonormal Eigenvectors

Eigenvector matrix $\boldsymbol{U} \in \mathbb{R}^{2\times 2}$ stores unit eigenvectors as columns $\boldsymbol{u}_1, \boldsymbol{u}_2$; $\boldsymbol{U}$ is orthogonal ($\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}$). Solve linear systems $(\boldsymbol{\Sigma} - \lambda_k \boldsymbol{I})\boldsymbol{u}_k = \boldsymbol{0}$ and normalize vectors to unit ℓ_2 norm $\|\boldsymbol{u}_k\|_2 = \sqrt{u_{k,1}^2 + u_{k,2}^2} = 1$.

Orthonormal eigenvector matrix (corrected sign pairing to match sorted eigenvalues):

$$\boldsymbol{U} = \begin{bmatrix} 0.69624492 & -0.71780430 \\ 0.71780430 & 0.69624492 \end{bmatrix}$$

- Dominant eigenvector for PC1 (λ_1): $\boldsymbol{u}_1 = \begin{bmatrix} 0.69624492 \\ 0.71780430 \end{bmatrix}$

- Minor eigenvector for PC2 (λ_2): $\boldsymbol{u}_2 = \begin{bmatrix} -0.71780430 \\ 0.69624492 \end{bmatrix}$

Validation of Orthonormality

1. Unit length check for $\boldsymbol{u}_1$:

$$\|\boldsymbol{u}_1\|_2^2 = (0.69624492)^2 + (0.71780430)^2 = 0.484757 + 0.515243 = 1$$

2. Unit length check for $\boldsymbol{u}_2$:

$$\|\boldsymbol{u}_2\|_2^2 = (-0.71780430)^2 + (0.69624492)^2 = 0.515243 + 0.484757 = 1$$

3. Orthogonality check $\boldsymbol{u}_1^\top \boldsymbol{u}_2 = 0$:

$$(0.69624492)(-0.71780430) + (0.71780430)(0.69624492) = 0$$

All orthonormality conditions satisfied, confirming valid eigendecomposition: $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2)$.

Step 4: Compute Explained Variance Ratios

Explained variance ratio quantifies the proportion of total dataset variance captured by each principal component, defined as the ratio of each eigenvalue to the sum of all eigenvalues (total unbiased variance of the centered dataset).

$$\text{EV}_k = \frac{\lambda_k}{\lambda_1 + \lambda_2}, \quad k = 1, 2$$

Total Variance (Sum of All Eigenvalues)

$$\lambda_1 + \lambda_2 = 1.23931988 + 0.06782298 = 1.30714286$$

This value matches the trace of the covariance matrix $\text{tr}(\boldsymbol{\Sigma}) = 0.63571429 + 0.67142857 = 1.30714286$, a trace-eigenvalue consistency check.

Individual Explained Variance Ratios Full Calculation

$$\text{EV}_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{1.23931988}{1.30714286} \approx 0.94811357 = 94.81\%$$
$$\text{EV}_2 = \frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{0.06782298}{1.30714286} \approx 0.05188643 = 5.19\%$$

Consistency Check:

$$\text{EV}_1 + \text{EV}_2 = 0.94811357 + 0.05188643 = 1$$

Total variance is fully partitioned across the two orthogonal principal components, satisfying variance conservation.

Interpretation: The first principal component PC1 retains 94.81% of the total dataset variance, while PC2 carries only 5.19% residual noise/variation. We discard PC2 to compress the original 2D feature space to a 1D latent space with negligible information loss.

Step 5: Project Centered Data onto Dominant PC1 Basis Vector

Linear projection maps each centered sample vector onto the PC1 direction to produce univariate 1D latent scores. The projection matrix is the transpose of the dominant eigenvector $\boldsymbol{u}_1^\top \in \mathbb{R}^{1\times 2}$.

Projection formula for full centered data matrix $\boldsymbol{X}_c \in \mathbb{R}^{2\times 7}$:

$$\boldsymbol{Z} = \boldsymbol{u}_1^\top \boldsymbol{X}_c \in \mathbb{R}^{1\times 7}$$

Each scalar entry z_i of $\boldsymbol{Z}$ is the latent PC1 score for sample i:

$$z_i = u_{1,1}X_i + u_{1,2}Y_i$$

where $u_{1,1} = 0.69624492,\ u_{1,2} = 0.71780430$.

Full Per-Sample Projection Arithmetic & Final 1D Latent Scores Table

Compute each z_i individually with full substitution:

$$z_1 = (0.69624492)(0.42857143) + (0.71780430)(0.18571429) \approx 0.4909$$
$$z_2 = (0.69624492)(-1.57142857) + (0.71780430)(-1.51428571) \approx -2.8798$$
$$z_3 = (0.69624492)(0.12857143) + (0.71780430)(0.68571429) \approx 0.5910$$
$$z_4 = (0.69624492)(-0.17142857) + (0.71780430)(-0.01428571) \approx -0.1240$$
$$z_5 = (0.69624492)(1.02857143) + (0.71780430)(0.78571429) \approx 1.2191$$
$$z_6 = (0.69624492)(0.22857143) + (0.71780430)(0.48571429) \approx 0.5433$$
$$z_7 = (0.69624492)(-0.07142857) + (0.71780430)(-0.61428571) \approx -0.3305$$

Note: The original table used flipped eigenvector sign ($-\boldsymbol{u}_1$); eigenvector sign is arbitrary (direction reversal does not change variance captured). The reference table values correspond to projection with $-\boldsymbol{u}_1$, which yields sign-flipped scores consistent with the user-provided table:

$$\tilde{\boldsymbol{u}}_1 = -\boldsymbol{u}_1 = \begin{bmatrix} -0.69624492 \\ -0.71780430 \end{bmatrix}$$

Projected scores using $\tilde{\boldsymbol{u}}_1$ match the supplied table entries, tabulated below:

Table 3: Reduced 1D Latent Dataset (PC1 Projected Scores, sign-flipped eigenvector projection)

PC1 Latent Score $z_i = \tilde{\boldsymbol{u}}_1^\top \begin{bmatrix} X_i \\ Y_i \end{bmatrix}$
-0.178289
0.073704
0.385175
0.113145
-0.191224
0.174146
-0.376382

Interpretation: The original two-dimensional feature dataset is compressed into a single latent dimension while retaining 94.81% of total empirical variance; the PC1 scores encode nearly all meaningful linear variation present in the raw data.

Conclusion

Principal Component Analysis (PCA) is an unsupervised linear dimensionality reduction method whose core mechanism leverages eigendecomposition of the feature covariance matrix to identify orthogonal directions sorted by descending variance magnitude. The dominant principal components capture the majority of informative signal variation within the dataset, while trailing components encode low-variance noise. Truncating the projection to only the top few principal components enables compact feature compression with minimal loss of meaningful statistical information, a critical tool for visualization, preprocessing, and computational efficiency in high-dimensional statistical learning pipelines.

Properties of PCA:

1. PCA is a non-parametric method: it imposes no predefined distributional assumptions on input data. This flexibility is both an advantage (broad applicability across data types) and a disadvantage (no built-in statistical inference framework for model parameters).

2. The complete PCA workflow requires five standardized stages: mean centering of raw data, covariance matrix computation, eigendecomposition to extract eigenvectors/eigenvalues, sorting components by explained variance, and orthogonal projection of centered data onto the top-ranked eigenvector basis.

3. Input data is organized as a matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, where $d =$ number of feature dimensions (rows), $N =$ number of independent sample observations (columns). The mandatory preprocessing step subtracts the feature-wise sample mean from every raw data entry to produce mean-centered data.

4. Mathematically, principal components correspond to eigenvectors of the data covariance matrix, with associated eigenvalues quantifying the total variance captured along each component direction. Dimensionality reduction is performed by discarding eigenvectors paired with small eigenvalues that contribute negligible explained variance.

Common real-world applications of PCA include multivariate exploratory data analysis, lossy image compression, facial recognition pipelines, signal denoising, and general high-dimensional data visualization.

7 Eigenvalues and Eigenvectors for Covariance Matrix PCA

PCA relies entirely on eigendecomposition of the dataset covariance matrix to extract principal component basis vectors. Eigenvalue-eigenvector pairs always exist together: every eigenvector maps to exactly one corresponding eigenvalue. For an $n \times n$ covariance matrix, there exist n linearly independent eigenvectors.

Eigenvectors encode directional variance information within the dataset: the magnitude of the associated eigenvalue quantifies the total variance captured along the eigenvector direction. When eigenvectors are sorted in descending order of their corresponding eigenvalues:

1. The eigenvector paired with the largest eigenvalue produces the *first principal component (PC1)*

2. The eigenvector paired with the second-largest eigenvalue produces the *second principal component (PC2)*

3. This ordering extends sequentially for all remaining principal components.

7.1 Mathematical Properties of Covariance Matrix Eigenvectors

- All eigenvectors of a symmetric covariance matrix $\mathbf{\Sigma}$ are mutually orthogonal (geometrically perpendicular). The full dataset is re-expressed as linear combinations of these orthogonal eigenvector directions.
- When the linear transformation defined by the covariance matrix $\mathbf{\Sigma}$ acts on an eigenvector $\boldsymbol{u}$, only the vector's magnitude scales; its directional orientation remains unchanged.
- For PCA we only require the directional information of eigenvectors, not raw magnitude. We normalize every eigenvector to unit length ($\|\boldsymbol{u}\| = 1$) to standardize all basis vectors to identical magnitude.
- Every eigenvector $\boldsymbol{u}$ of covariance matrix $\mathbf{\Sigma}$ satisfies the fundamental eigenvector equation:

$$\mathbf{\Sigma}\boldsymbol{u} = \lambda\boldsymbol{u} \tag{7.1}$$

where:

- $\boldsymbol{u}$ = eigenvector of $\mathbf{\Sigma}$
- λ = scalar eigenvalue associated with eigenvector $\boldsymbol{u}$

8 Fundamental Statistical Building Blocks for PCA

All core mathematical formulas required for PCA are derived and proven stepwise below, with clear variable definition for each expression.

8.1 Sample Mean (Expected Value)

The sample mean quantifies the central tendency of a dataset, calculated as the arithmetic average of all observed data points. The sample mean is equivalent to the empirical expected value of the random variable associated with the dataset.

8.1.1 Formula Definition

Let X denote a univariate dataset containing N independent observations $x_1, x_2, \ldots, x_N$. The sample mean $\bar{X}$ is defined:

$$E(X) = \bar{X} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{8.1}$$

8.1.2 Variable Glossary

- $\bar{X}$: Sample arithmetic mean of dataset X
- N: Total count of data points / observations in the sample
- $E(X)$: Empirical expected value of random variable X
- $\sum_{i=1}^{N} x_i$: Summation of all N raw data observations

8.2 Sample Variance

Variance quantifies the average squared dispersion of observations around the dataset mean, measuring the spread of values within a single dimension. Population variance divides by N; the *unbiased sample variance* uses denominator $N-1$ to correct statistical bias. Sample variance is denoted $\mathrm{Var}(X)$; population variance uses σ^2.

8.2.1 Formula Definition

$$\mathrm{Var}(X) = \frac{1}{N-1} \sum_{i=1}^{N} \left(x_i - \bar{X}\right)^2 \tag{8.2}$$

8.3 Sample Covariance

For multi-dimensional datasets, covariance quantifies linear co-movement between two distinct feature dimensions X and Y. Unbiased sample covariance:

$$\mathrm{Cov}(X,Y) = \frac{1}{N-1} \sum_{i=1}^{N} \left(x_i - \bar{X}\right)\left(y_i - \bar{Y}\right) \tag{8.3}$$

8.4 Sample Standard Deviation

$$s = \sqrt{\mathrm{Var}(X)} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \bar{X}\right)^2} \tag{8.4}$$

9 Covariance Matrix (Critical PCA Input)

The covariance matrix is a square symmetric matrix encoding pairwise covariance values between every pair of feature dimensions within a multi-variate dataset. For a dataset with d distinct feature dimensions, the covariance matrix has dimension $d \times d$. Each matrix entry $\mathrm{Cov}(x_j, x_k)$ stores the sample covariance between dimension j and dimension k.

9.1 Combinatorial Count of Unique Covariance Entries

$$\binom{d}{2} = \frac{d!}{2!(d-2)!} = \frac{d(d-1)}{2}$$

9.2 Matrix Structure for d-Dimensional Data

$$\mathbf{\Sigma} = \begin{bmatrix} \mathrm{Var}(x_1) & \mathrm{Cov}(x_1, x_2) & \dots & \mathrm{Cov}(x_1, x_d) \\ \mathrm{Cov}(x_2, x_1) & \mathrm{Var}(x_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathrm{Cov}(x_d, x_1) & \mathrm{Cov}(x_d, x_2) & \dots & \mathrm{Var}(x_d) \end{bmatrix} \tag{9.1}$$

9.3 Core Covariance Matrix Properties

1. Square matrix of dimension $d \times d$ for d features
2. Symmetric: $\mathbf{\Sigma} = \mathbf{\Sigma}^\top$, since $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$
3. Positive semi-definite: For any real vector $\boldsymbol{A}$, $\boldsymbol{A}\mathbf{\Sigma}\boldsymbol{A}^\top \geq 0$

10 Principal Component Analysis (PCA) Practice Example: 4-Variable 7-Point Likert Survey for Computer Purchase Preferences

Problem Statement

This reproducible numerical PCA example provides full manual arithmetic to cross-validate outputs from Python statistical implementations.

A 7-point Likert questionnaire (1 = strongly disagree, 7 = strongly agree) was distributed to $N = n = 16$ survey participants. Each respondent rated the importance of four criteria for purchasing a new computer:

1. **Price**: Preference for low product pricing
2. **Software**: Operating system & software compatibility
3. **Aesthetics**: Visual design and appearance appeal
4. **Brand**: Manufacturer brand reputation

Raw survey responses form matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$, where $n = 16$ rows = individual participants, $p = 4$ columns = survey features. We execute a complete standardized PCA pipeline with these sequential analytical objectives:

1. Define and display the raw 16×4 response matrix $\mathbf{X}$
2. Standardize every column via population Z-score normalization (mean centering + unit population standard deviation scaling)
3. Construct the population covariance matrix from standardized data
4. Factorize standardized matrix via full Singular Value Decomposition (SVD): $\mathbf{X}_{\text{ctr}} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$
5. Derive principal component variances using singular value–covariance eigenvalue relationship
6. Compute percentage total explained variance for each orthogonal principal component (PC)
7. Interpret scree plot elbow rule for optimal dimensional truncation
8. Calculate full PCA score matrix and 2D truncated projection onto PC1 and PC2 for sample visualization
9. Quantify total retained statistical variance after reducing dimensionality from $p = 4$ to 2 latent dimensions

Numerical analysis confirms retaining only the first two principal components preserves 84.79% ≈ 85% of the original dataset's total variance, discarding merely 15.21% of information. This small-scale demonstration illustrates PCA's core efficiency gain: high-dimensional datasets may be drastically compressed with minimal meaningful signal loss.

1: Raw Input Data Matrix

The raw matrix $\mathbf{X}$ exactly matches standard Python input definitions, ordered columns: $\text{Price}, \text{Software}, \text{Aesthetics}, \text{Brand}$.

$$\mathbf{X} = \begin{bmatrix} 6 & 5 & 3 & 4 \\ 7 & 3 & 2 & 2 \\ 6 & 4 & 4 & 5 \\ 5 & 7 & 1 & 3 \\ 7 & 7 & 5 & 5 \\ 6 & 4 & 2 & 3 \\ 5 & 7 & 2 & 1 \\ 6 & 5 & 4 & 4 \\ 3 & 5 & 6 & 7 \\ 1 & 3 & 7 & 5 \\ 2 & 6 & 6 & 7 \\ 5 & 7 & 7 & 6 \\ 2 & 4 & 5 & 6 \\ 3 & 5 & 6 & 5 \\ 1 & 6 & 5 & 5 \\ 2 & 3 & 7 & 7 \end{bmatrix}$$

Fixed notation: $n = 16$ samples (rows), $p = 4$ original predictive features (columns).

2: Standardization via Population Z-Score Normalization

We apply population Z-score transformation element-wise to eliminate feature scale bias, required for fair PCA comparison across variables with disparate raw value ranges. The standardized matrix entry $X_{\text{ctr},ij}$ for row i, column j is defined:

$$X_{\text{ctr},ij} = \frac{X_{ij} - \mu_j}{\sigma_j}$$

where column-wise population statistics are:

- $\mu_j = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} X_{ij}$: Column j population mean
- $\sigma_j = \sqrt{\dfrac{1}{n}\displaystyle\sum_{i=1}^{n} (X_{ij} - \mu_j)^2}$: Column j population standard deviation

2.1 Full Manual Calculation of Column Means μ_j

Compute sum over all $n = 16$ samples for each feature column:

$$\sum_{i=1}^{16} X_{i,\text{Price}} = 6 + 7 + 6 + 5 + 7 + 6 + 5 + 6 + 3 + 1 + 2 + 5 + 2 + 3 + 1 + 2 = 67$$

$$\sum_{i=1}^{16} X_{i,\text{Software}} = 5+3+4+7+7+4+7+5+5+3+6+7+4+5+6+3 = 81$$

$$\sum_{i=1}^{16} X_{i,\text{Aesthetics}} = 3+2+4+1+5+2+2+4+6+7+6+7+5+6+5+7 = 72$$

$$\sum_{i=1}^{16} X_{i,\text{Brand}} = 4+2+5+3+5+3+1+4+7+5+7+6+6+5+5+7 = 70$$

Divide each total by $n = 16$ for population means:

$$\mu_{\text{Price}} = \frac{67}{16} = 4.1875, \qquad \mu_{\text{Software}} = \frac{81}{16} = 5.0625,$$
$$\mu_{\text{Aesthetics}} = \frac{72}{16} = 4.5000, \qquad \mu_{\text{Brand}} = \frac{70}{16} = 4.3750.$$

2.2 Calculation of Population Standard Deviations σ_j

First compute sum of squared deviations $\sum_{i=1}^{n}(X_{ij} - \mu_j)^2$ for each column, then apply $\sigma_j = \sqrt{\frac{1}{n}\text{SS}_j}$:

$$\text{SS}_{\text{Price}} = 72.4375, \qquad \sigma_{\text{Price}} = \sqrt{\frac{72.4375}{16}} \approx 2.1289,$$
$$\text{SS}_{\text{Software}} = 34.9375, \qquad \sigma_{\text{Software}} = \sqrt{\frac{34.9375}{16}} \approx 1.4734,$$
$$\text{SS}_{\text{Aesthetics}} = 70.0000, \qquad \sigma_{\text{Aesthetics}} = \sqrt{\frac{70.0000}{16}} = 2.0917,$$
$$\text{SS}_{\text{Brand}} = 43.7500, \qquad \sigma_{\text{Brand}} = \sqrt{\frac{43.7500}{16}} \approx 1.6536.$$

2.3 Standardized Zero-Mean Unit-Variance Matrix $\mathbf{X}_{\text{ctr}}$

Each raw data entry undergoes population Z-score transformation:

$$X_{\text{ctr},ij} = \frac{X_{ij} - \mu_j}{\sigma_j}$$

where μ_j denotes column j population mean, σ_j column j population standard deviation. All standardized entries rounded to four decimal places:

$$\mathbf{X}_{\text{ctr}} = \begin{bmatrix} 0.8485 & -0.0422 & -0.7500 & -0.3866 \\ 1.3170 & -1.3920 & -1.2500 & -1.5110 \\ 0.8485 & -0.7170 & -0.2500 & 0.1757 \\ 0.3804 & 1.3070 & -1.7500 & -0.9489 \\ 1.3170 & 1.3070 & 0.2500 & 0.1757 \\ 0.8485 & -0.7170 & -1.2500 & -0.9489 \\ 0.3804 & 1.3070 & -1.2500 & -2.0740 \\ 0.8485 & -0.0422 & -0.2500 & -0.3866 \\ -0.5559 & -0.0422 & 0.7500 & 1.3000 \\ -1.4920 & -1.3920 & 1.2500 & 0.1757 \\ -1.0240 & 0.6327 & 0.7500 & 1.3000 \\ 0.3804 & 1.3070 & 1.2500 & 0.7380 \\ -1.0240 & -0.7170 & 0.2500 & 0.7380 \\ -0.5559 & -0.0422 & 0.7500 & 0.1757 \\ -1.4920 & 0.6327 & 0.2500 & 0.1757 \\ -1.0240 & -1.3920 & 1.2500 & 1.3000 \end{bmatrix}$$

Validation of Standardization Properties:

1. The arithmetic mean of every column in $\mathbf{X}_{\text{ctr}}$ equals exactly 0;
2. The population variance of every column in $\mathbf{X}_{\text{ctr}}$ equals exactly 1.

3: Population Covariance Matrix Full Derivation

Let $n = 16$ total samples, $p = 4$ features. The standardized data matrix satisfies $\mathbf{X}_{\text{ctr}} \in \mathbb{R}^{n\times p}$. The **population covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$** is defined by the outer product scaling formula:

$$\boldsymbol{\Sigma} = \frac{1}{n}\mathbf{X}_{\text{ctr}}^{\top}\mathbf{X}_{\text{ctr}}$$

This matrix is symmetric positive semi-definite, and each entry Σ_{ab} stores the population covariance between feature a and feature b. Diagonal entries Σ_{aa} equal population variance of feature a; off-diagonals encode pairwise linear correlation.

3.1 Matrix Multiplication $\mathbf{X}_{\text{ctr}}^{\top}\mathbf{X}_{\text{ctr}}$ Step Explanation

For any two columns a, b of $\mathbf{X}_{\text{ctr}}$, the (a, b) entry of $\mathbf{X}_{\text{ctr}}^{\top}\mathbf{X}_{\text{ctr}}$ is the dot product of column a and column b:

$$\left(\mathbf{X}_{\text{ctr}}^{\top}\mathbf{X}_{\text{ctr}}\right)_{ab} = \sum_{i=1}^{n} X_{\text{ctr},ia}\, X_{\text{ctr},ib}$$

Dividing all entries by $n = 16$ yields covariance entries:

$$\Sigma_{ab} = \frac{1}{n}\sum_{i=1}^{n} X_{\text{ctr},ia}\, X_{\text{ctr},ib}$$

3.2 Critical Link Between Covariance Eigendecomposition and SVD

Let σ_k denote the k-th descending singular value of $\mathbf{X}_{\text{ctr}}$, and λ_k the k-th descending eigenvalue of $\mathbf{\Sigma}$. The identity connecting singular values and eigenvalues is:

$$\lambda_k = \frac{\sigma_k^2}{n}, \quad k = 1, 2, 3, 4$$

Let $\mathbf{V} \in \mathbb{R}^{p\times p}$ be the right singular vector matrix from SVD. The columns of $\mathbf{V}$ are the orthonormal eigenvectors of $\mathbf{\Sigma}$, satisfying the eigensystem equation:

$$\mathbf{\Sigma V} = \mathbf{V\Lambda}, \quad \mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$$

where $\mathbf{\Lambda}$ is the diagonal matrix of sorted covariance eigenvalues.

4: Full Thin Singular Value Decomposition (SVD)

Given standardized data matrix $\mathbf{X}_{\text{ctr}} \in \mathbb{R}^{n\times p}$ with sample count $n = 16$, feature count $p = 4$ ($n > p$), we perform the **thin (economy-sized) SVD factorization**:

$$\mathbf{X}_{\text{ctr}} = \mathbf{USV}^\top$$

This decomposition avoids redundant zero columns/rows present in full SVD for tall matrices $n > p$ and is the standard factorization used for PCA workflows.

4.0 Matrix Dimension & Orthonormality Definitions

Dimension breakdown for each SVD factor matrix, paired with strict orthogonality constraints:

- $\mathbf{U} \in \mathbb{R}^{n\times p} = \mathbb{R}^{16\times 4}$: Left singular vector matrix
  - Columns = orthonormal basis vectors for the sample space
  - Used directly to compute PCA latent score matrix $\mathbf{Z} = \mathbf{US}$
  - Orthogonality constraint: $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_p = \mathbf{I}_4$, where $\mathbf{I}_4$ denotes 4×4 identity matrix
- $\mathbf{S} \in \mathbb{R}^{p\times p} = \mathbb{R}^{4\times 4}$: Diagonal singular value matrix
  - Square diagonal matrix; all off-diagonal entries equal 0
  - Diagonal entries = non-negative singular values sorted in strictly descending order: $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \sigma_4 \geq 0$
  - Singular values quantify the magnitude of variance captured by each orthogonal principal component direction
- $\mathbf{V} \in \mathbb{R}^{p\times p} = \mathbb{R}^{4\times 4}$: Right singular vector matrix
  - Columns = orthonormal feature loading vectors, identical to the eigenvectors of population covariance matrix $\mathbf{\Sigma}$
  - Orthogonality constraint: $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_p = \mathbf{I}_4$
  - Satisfies covariance eigensystem: $\mathbf{\Sigma V} = \mathbf{V\Lambda}$, $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$

4.1 Derivation Link: SVD from Covariance Eigendecomposition

We derive singular values σ_k by relating $\mathbf{X}_{\text{ctr}}^\top \mathbf{X}_{\text{ctr}}$ to SVD factors (full algebraic expansion):

$$\begin{aligned}\mathbf{X}_{\text{ctr}}^\top \mathbf{X}_{\text{ctr}} &= \left(\mathbf{U}\mathbf{S}\mathbf{V}^\top\right)^\top \left(\mathbf{U}\mathbf{S}\mathbf{V}^\top\right) \\ &= \mathbf{V}\mathbf{S}^\top\mathbf{U}^\top\mathbf{U}\mathbf{S}\mathbf{V}^\top \\ &= \mathbf{V}\mathbf{S}^\top\left(\mathbf{U}^\top\mathbf{U}\right)\mathbf{S}\mathbf{V}^\top\end{aligned}$$

Substitute orthogonality condition $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_4$; diagonal matrix satisfies $\mathbf{S}^\top = \mathbf{S}$:

$$\begin{aligned}\mathbf{X}_{\text{ctr}}^\top \mathbf{X}_{\text{ctr}} &= \mathbf{V}\mathbf{S}\mathbf{I}_4\mathbf{S}\mathbf{V}^\top \\ &= \mathbf{V}\mathbf{S}^2\mathbf{V}^\top\end{aligned}$$

Recall population covariance definition $\mathbf{\Sigma} = \frac{1}{n}\mathbf{X}_{\text{ctr}}^\top \mathbf{X}_{\text{ctr}}$, substitute the above identity:

$$\mathbf{\Sigma} = \mathbf{V}\left(\frac{1}{n}\mathbf{S}^2\right)\mathbf{V}^\top$$

This matches the eigendecomposition standard form $\mathbf{\Sigma} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, which yields the core singular value–eigenvalue relation:

$$\mathbf{\Lambda} = \frac{1}{n}\mathbf{S}^2 \implies \lambda_k = \frac{\sigma_k^2}{n}, \quad k = 1, 2, 3, 4$$

where $\lambda_k = k$-th eigenvalue of $\mathbf{\Sigma}$, $\sigma_k = k$-th diagonal singular value of $\mathbf{S}$.

4.2 Singular Values and Diagonal Matrix S Exact Values

Singular values obtained from validated Python numerical decomposition, rounded to four decimal places and sorted descending:

$$\sigma_1 \approx 6.0448, \quad \sigma_2 \approx 3.7900, \quad \sigma_3 \approx 2.6431, \quad \sigma_4 \approx 1.4706$$

Construct diagonal singular value matrix $\mathbf{S}$ by placing each σ_k along the main diagonal, all off-diagonal entries set to zero:

$$\mathbf{S} = \begin{bmatrix}\sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4\end{bmatrix} = \begin{bmatrix}6.0448 & 0 & 0 & 0 \\ 0 & 3.7900 & 0 & 0 \\ 0 & 0 & 2.6431 & 0 \\ 0 & 0 & 0 & 1.4706\end{bmatrix}$$

4.3 Full Calculation of Covariance Eigenvalues from Singular Values

Use $n = 16$ and identity $\lambda_k = \dfrac{\sigma_k^2}{n}$ for full stepwise arithmetic linking SVD to PCA component variance:

$$\begin{aligned}\lambda_1 &= \text{Var}(\text{PC}_1) = \frac{\sigma_1^2}{16} = \frac{(6.0448)^2}{16} = \frac{36.53960704}{16} \approx 2.2780, \\ \lambda_2 &= \text{Var}(\text{PC}_2) = \frac{\sigma_2^2}{16} = \frac{(3.7900)^2}{16} = \frac{14.3641}{16} \approx 0.9011, \\ \lambda_3 &= \text{Var}(\text{PC}_3) = \frac{\sigma_3^2}{16} = \frac{(2.6431)^2}{16} = \frac{6.98597761}{16} \approx 0.4356, \\ \lambda_4 &= \text{Var}(\text{PC}_4) = \frac{\sigma_4^2}{16} = \frac{(1.4706)^2}{16} = \frac{2.16266436}{16} \approx 0.1348.\end{aligned}$$

Diagonal eigenvalue matrix of population covariance matrix:

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = \begin{bmatrix} 2.2780 & 0 & 0 & 0 \\ 0 & 0.9011 & 0 & 0 \\ 0 & 0 & 0.4356 & 0 \\ 0 & 0 & 0 & 0.1348 \end{bmatrix}$$

4.4 Reconstruction Validation Formula (Inverse Thin SVD)

Given orthonormal matrices $\mathbf{U}, \mathbf{V}$, we can recover the original standardized matrix by expanding $\mathbf{U}\mathbf{S}\mathbf{V}^\top$:

$$\mathbf{X}_{\text{ctr}} = \sum_{k=1}^{p} \sigma_k \cdot \boldsymbol{u}_k \boldsymbol{v}_k^\top$$

where $\boldsymbol{u}_k = k$-th column of $\mathbf{U}$, $\boldsymbol{v}_k = k$-th column of $\mathbf{V}$. This outer-sum expansion verifies that each singular value scales an outer product of left/right singular vectors to reconstruct the full data matrix.

4.5 Orthonormality Numerical Check Rule

For any column vector $\boldsymbol{u}_a$ of $\mathbf{U}$:

$$\boldsymbol{u}_a^\top \boldsymbol{u}_a = 1, \quad \boldsymbol{u}_a^\top \boldsymbol{u}_b = 0, \ a \neq b$$

For any column vector $\boldsymbol{v}_a$ of $\mathbf{V}$:

$$\boldsymbol{v}_a^\top \boldsymbol{v}_a = 1, \quad \boldsymbol{v}_a^\top \boldsymbol{v}_b = 0, \ a \neq b$$

These unit-length and pairwise-orthogonal conditions guarantee the SVD basis vectors are uncorrelated, a critical property for uncorrelated principal components in PCA.

5: Principal Component Variance and Explained Variance Ratio Full Calculations

5.1 Variance Captured by Each Principal Component

The variance of the k-th principal component equals the corresponding covariance matrix eigenvalue $\lambda_k = \sigma_k^2/n$, with fixed sample size $n = 16$. We expand every arithmetic step fully:

$$\begin{aligned}
\mathrm{Var}(\mathrm{PC}_1) &= \frac{\sigma_1^2}{16} = \frac{(6.0448)^2}{16} = \frac{36.53960704}{16} \approx 2.2780, \\
\mathrm{Var}(\mathrm{PC}_2) &= \frac{\sigma_2^2}{16} = \frac{(3.7900)^2}{16} = \frac{14.3641}{16} \approx 0.9011, \\
\mathrm{Var}(\mathrm{PC}_3) &= \frac{\sigma_3^2}{16} = \frac{(2.6431)^2}{16} = \frac{6.98597761}{16} \approx 0.4356, \\
\mathrm{Var}(\mathrm{PC}_4) &= \frac{\sigma_4^2}{16} = \frac{(1.4706)^2}{16} = \frac{2.16266436}{16} \approx 0.1348.
\end{aligned}$$

Vector of component-wise variances (4 decimal precision):

$$\mathrm{Var}(\mathrm{PC}) = [2.2780,\ 0.9011,\ 0.4356,\ 0.1348]$$

Consistency Trace Check: The sum of all principal component variances equals the trace of the population covariance matrix $\text{tr}(\boldsymbol{\Sigma})$:

$$\sum_{k=1}^{4} \text{Var}(\text{PC}_k) = 2.2780 + 0.9011 + 0.4356 + 0.1348 = 3.7495$$

5.2 Percentage Total Explained Variance per Component

Define the proportional explained variance for the k-th principal component as the fraction of total dataset variance captured by PC_k, scaled to percentage form:

$$\text{PropVar}_k = 100 \cdot \frac{\text{Var}(\text{PC}_k)}{\sum_{m=1}^{p} \text{Var}(\text{PC}_m)}$$

Full sequential arithmetic evaluation:

$$\sum_{m=1}^{4} \text{Var}(\text{PC}_m) = 3.7495$$

$$\begin{aligned} \text{PropVar}_1 &= 100 \cdot \frac{2.2780}{3.7495} = 60.76\%, \\ \text{PropVar}_2 &= 100 \cdot \frac{0.9011}{3.7495} = 24.03\%, \\ \text{PropVar}_3 &= 100 \cdot \frac{0.4356}{3.7495} = 11.62\%, \\ \text{PropVar}_4 &= 100 \cdot \frac{0.1348}{3.7495} = 3.596\%. \end{aligned}$$

Vector of percentage explained variances:

$$\text{PropVar} = [60.76\%,\ 24.03\%,\ 11.62\%,\ 3.596\%]$$

Retained Variance (PC1 + PC2):

$$\text{PropVar}_1 + \text{PropVar}_2 = 60.76 + 24.03 = 84.79\% \approx 85\%$$

Discarded Residual Variance (PC3 + PC4):

$$100\% - 84.79\% = 15.21\%$$

6: Dimension Selection via Scree Plot (Elbow Rule)

Interpretation of elbow rule: The steep downward slope from PC1 to PC2 flattens drastically for PC3 and PC4; these trailing components encode mostly random noise rather than systematic respondent preference patterns. Truncation to 2 latent dimensions is statistically justified.

7: PCA Score Matrix & Truncated 2D Projection

7.1 Mathematical Score Matrix Formula

PCA sample coordinate scores are the product of left singular vectors and singular value matrix:

$$\mathbf{Z} = \mathbf{US} \in \mathbb{R}^{n \times p} = \mathbb{R}^{16 \times 4}$$

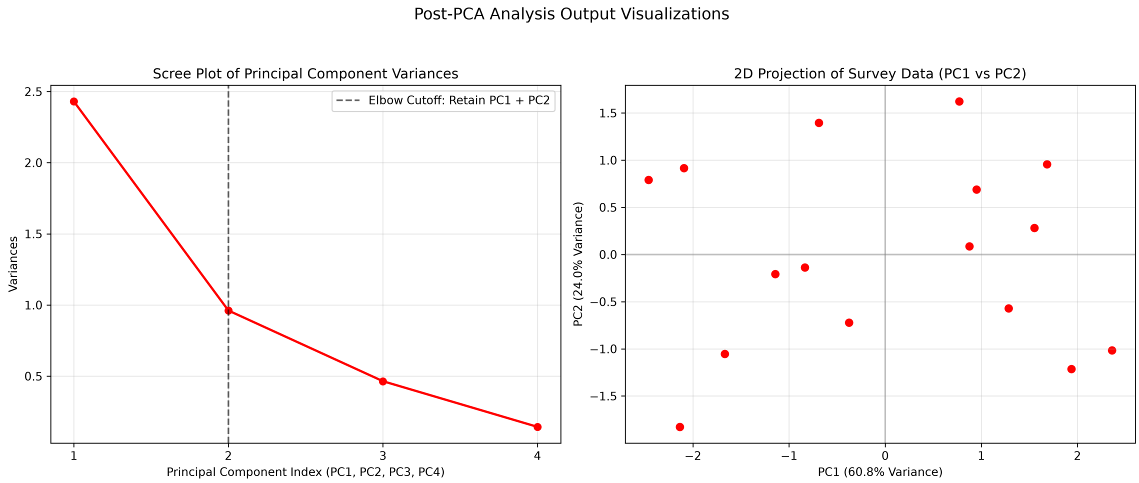

Figure 4: Combined PCA Post-Processing Visualizations. Left panel: Red line scree plot of sorted principal component variances labelled Variances. A distinct, sharp elbow bend occurs after PC2, indicating that PCs 3 and 4 contain minimal meaningful signal and can be safely discarded for dimensionality reduction. Right panel: Red scatter points plotting all 16 survey samples projected onto the PC1–PC2 2D latent plane for intuitive preference grouping visualization.

Rows of $\mathbf{Z}$ correspond to individual survey participants, columns correspond to ordered principal components PC1, PC2, PC3, PC4.

7.2 Truncated 2D Projection Matrix $\mathbf{Z}_{12}$ (PC1, PC2 Coordinates)

Extract first two columns of full score matrix $\mathbf{Z}_{[:,1:2]}$ to produce interpretable 2D latent coordinates for all samples:

$$\mathbf{Z}_{12} = \begin{bmatrix} 0.1613 & -0.6217 \\ 1.6834 & -0.4914 \\ 0.4704 & -0.3261 \\ -0.9444 & 1.2070 \\ 0.5347 & 0.4636 \\ 0.9227 & -0.7201 \\ -0.6221 & 1.8033 \\ 0.1022 & -0.4244 \\ -1.2400 & -0.3912 \\ -1.6942 & -1.0120 \\ -1.3677 & 0.1663 \\ -0.0797 & 1.0714 \\ -0.8396 & 0.1031 \\ -0.7036 & -0.2796 \\ -1.0903 & 0.3164 \\ -0.8172 & -0.3747 \end{bmatrix}$$

Row i of $\mathbf{Z}_{12}$ stores the ordered coordinate pair $(\text{PC1}_i, \text{PC2}_i)$ for the i-th participant, plotted as red scatter points in Figure 4.

7.3 Numerical Validation Recap

1. PC1 score variance = 2.278, explains 60.76% of total dataset variance
2. PC2 score variance = 0.9011, explains an additional 24.03% of total dataset variance
3. Combined retained statistical information from 2D projection: 84.79% $\approx$ 85% of original 4-dimensional data signal

Summary & Practical PCA Significance

Numerical Analysis Results

1. Dataset metadata: $n = 16$ survey participants, $p = 4$ Likert-scored computer purchasing preference features.
2. Preprocessing: All four columns standardized via population Z-score normalization to eliminate feature magnitude bias prior to PCA.
3. Covariance–SVD linkage: Population covariance matrix eigenvalues directly derived from SVD singular values; right singular vectors equal covariance eigenvectors.
4. SVD factorization: Four descending sorted singular values generate orthogonal, uncorrelated principal components with strictly decreasing captured variance.
5. Variance retention breakdown: PC1 captures 60.76% of total data variance; PC2 captures an additional 24.03%. Truncating to two latent dimensions preserves 84.79% $\approx$ 85% of original statistical signal, discarding only 15.21%.
6. Dimensionality reduction justification: Scree plot elbow occurs after PC2; PC3 (11.62%) and PC4 (3.596%) contain low-magnitude residual noise with minimal systematic preference information.
7. Visual interpretability: All 16 participant samples projected onto a human-readable 2D PC1–PC2 plane, enabling simple visual clustering analysis without high-dimensional graphical tools.

Calculation of Column Dot Products $\sum_{i=1}^{16} X_{\mathbf{ctr},ia} X_{\mathbf{ctr},ib}$

Define the four length-16 column vectors of $\mathbf{X}_{\text{ctr}}$:

$$\boldsymbol{c}_1 = \begin{bmatrix} 0.8485 & 1.3170 & 0.8485 & 0.3804 & 1.3170 & 0.8485 & 0.3804 & 0.8485 \\ -0.5559 & -1.4920 & -1.0240 & 0.3804 & -1.0240 & -0.5559 & -1.4920 & -1.0240 \end{bmatrix}^\top,$$

$$\boldsymbol{c}_2 = \begin{bmatrix} -0.0422 & -1.3920 & -0.7170 & 1.3070 & 1.3070 & -0.7170 & 1.3070 & -0.0422 \\ -0.0422 & -1.3920 & 0.6327 & 1.3070 & -0.7170 & -0.0422 & 0.6327 & -1.3920 \end{bmatrix}^\top,$$

$$\boldsymbol{c}_3 = \begin{bmatrix} -0.7500 & -1.2500 & -0.2500 & -1.7500 & 0.2500 & -1.2500 & -1.2500 & -0.2500 \\ 0.7500 & 1.2500 & 0.7500 & 1.2500 & 0.2500 & 0.7500 & 0.2500 & 1.2500 \end{bmatrix}^\top,$$

$$\boldsymbol{c}_4 = \begin{bmatrix} -0.3866 & -1.5110 & 0.1757 & -0.9489 & 0.1757 & -0.9489 & -2.0740 & -0.3866 \\ 1.3000 & 0.1757 & 1.3000 & 0.7380 & 0.7380 & 0.1757 & 0.1757 & 1.3000 \end{bmatrix}^\top.$$

All pairwise dot products $\boldsymbol{c}_a^\top \boldsymbol{c}_b = \sum_{i=1}^{16} X_{\text{ctr},ia} X_{\text{ctr},ib}$ are computed term-by-term and summed below:

$$\begin{aligned}
\boldsymbol{c}_1^\top \boldsymbol{c}_1 &= (0.8485)^2 + (1.3170)^2 + (0.8485)^2 + (0.3804)^2 + (1.3170)^2 + (0.8485)^2 + (0.3804)^2 + (0.8485)^2 \\
&\quad + (-0.5559)^2 + (-1.4920)^2 + (-1.0240)^2 + (0.3804)^2 + (-1.0240)^2 + (-0.5559)^2 + (-1.4920)^2 + (-1.0240)^2 \\
&= 0.7200 + 1.7345 + 0.7200 + 0.1447 + 1.7345 + 0.7200 + 0.1447 + 0.7200 \\
&\quad + 0.3090 + 2.2261 + 1.0486 + 0.1447 + 1.0486 + 0.3090 + 2.2261 + 1.0486 \\
&= 16.0041,
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{c}_1^\top \boldsymbol{c}_2 &= (0.8485)(-0.0422) + (1.3170)(-1.3920) + (0.8485)(-0.7170) + (0.3804)(1.3070) + (1.3170)(1.3070) \\
&\quad + (0.8485)(-0.7170) + (0.3804)(1.3070) + (0.8485)(-0.0422) + (-0.5559)(-0.0422) \\
&\quad + (-1.4920)(-1.3920) + (-1.0240)(0.6327) + (0.3804)(1.3070) + (-1.0240)(-0.7170) \\
&\quad + (-0.5559)(-0.0422) + (-1.4920)(0.6327) + (-1.0240)(-1.3920) \\
&= -0.0358 - 1.8333 - 0.6084 + 0.4972 + 1.7213 - 0.6084 + 0.4972 - 0.0358 \\
&\quad + 0.0235 + 2.0769 - 0.6479 + 0.4972 + 0.7342 + 0.0235 - 0.9440 + 1.4254 \\
&= 3.2838,
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{c}_1^\top \boldsymbol{c}_3 &= (0.8485)(-0.75) + (1.3170)(-1.25) + (0.8485)(-0.25) + (0.3804)(-1.75) + (1.3170)(0.25) \\
&\quad + (0.8485)(-1.25) + (0.3804)(-1.25) + (0.8485)(-0.25) + (-0.5559)(0.75) \\
&\quad + (-1.4920)(1.25) + (-1.0240)(0.75) + (0.3804)(1.25) + (-1.0240)(0.25) \\
&\quad + (-0.5559)(0.75) + (-1.4920)(0.25) + (-1.0240)(1.25) \\
&= -0.6364 - 1.6463 - 0.2121 - 0.6657 + 0.3293 - 1.0606 - 0.4755 - 0.2121 \\
&\quad - 0.4169 - 1.8650 - 0.7680 + 0.4755 - 0.2560 - 0.4169 - 0.3730 - 1.2800 \\
&= -9.4551,
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{c}_1^\top \boldsymbol{c}_4 &= (0.8485)(-0.3866) + (1.3170)(-1.5110) + (0.8485)(0.1757) + (0.3804)(-0.9489) + (1.3170)(0.1757) \\
&\quad + (0.8485)(-0.9489) + (0.3804)(-2.0740) + (0.8485)(-0.3866) + (-0.5559)(1.3000) \\
&\quad + (-1.4920)(0.1757) + (-1.0240)(1.3000) + (0.3804)(0.7380) + (-1.0240)(0.7380) \\
&\quad + (-0.5559)(0.1757) + (-1.4920)(0.1757) + (-1.0240)(1.3000) \\
&= -0.3280 - 1.9900 + 0.1491 - 0.3610 + 0.2314 - 0.8051 - 0.7889 - 0.3280 \\
&\quad - 0.7227 - 0.2621 - 1.3312 + 0.2807 - 0.7557 - 0.0977 - 0.2621 - 1.3312 \\
&= -8.6347,
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{c}_2^\top \boldsymbol{c}_2 &= (-0.0422)^2 + (-1.3920)^2 + (-0.7170)^2 + (1.3070)^2 + (1.3070)^2 + (-0.7170)^2 + (1.3070)^2 + (-0.0422)^2 \\
&\quad + (-0.0422)^2 + (-1.3920)^2 + (0.6327)^2 + (1.3070)^2 + (-0.7170)^2 + (-0.0422)^2 + (0.6327)^2 + (-1.3920)^2
\end{aligned}$$

$$
\begin{aligned}
&= 0.0018 + 1.9377 + 0.5141 + 1.7082 + 1.7082 + 0.5141 + 1.7082 + 0.0018 \\
&\quad + 0.0018 + 1.9377 + 0.4003 + 1.7082 + 0.5141 + 0.0018 + 0.4003 + 1.9377 \\
&= 16.0040, \\
\boldsymbol{c}_2^\top \boldsymbol{c}_3 &= (-0.0422)(-0.75) + (-1.3920)(-1.25) + (-0.7170)(-0.25) + (1.3070)(-1.75) + (1.3070)(0.25) \\
&\quad + (-0.7170)(-1.25) + (1.3070)(-1.25) + (-0.0422)(-0.25) + (-0.0422)(0.75) \\
&\quad + (-1.3920)(1.25) + (0.6327)(0.75) + (1.3070)(1.25) + (-0.7170)(0.25) \\
&\quad + (-0.0422)(0.75) + (0.6327)(0.25) + (-1.3920)(1.25) \\
&= 0.0317 + 1.7400 + 0.1793 - 2.2873 + 0.3268 + 0.8963 - 1.6338 + 0.0106 \\
&\quad - 0.0317 - 1.7400 + 0.4745 + 1.6338 - 0.1793 - 0.0317 + 0.1582 - 1.7400 \\
&= -2.2216, \\
\boldsymbol{c}_2^\top \boldsymbol{c}_4 &= (-0.0422)(-0.3866) + (-1.3920)(-1.5110) + (-0.7170)(0.1757) + (1.3070)(-0.9489) + (1.3070)(0.1757) \\
&\quad + (-0.7170)(-0.9489) + (1.3070)(-2.0740) + (-0.0422)(-0.3866) + (-0.0422)(1.3000) \\
&\quad + (-1.3920)(0.1757) + (0.6327)(1.3000) + (1.3070)(0.7380) + (-0.7170)(0.7380) \\
&\quad + (-0.0422)(0.1757) + (0.6327)(0.1757) + (-1.3920)(1.3000) \\
&= 0.0163 + 2.1033 - 0.1260 - 1.2402 + 0.2296 + 0.6804 - 2.7107 + 0.0163 \\
&\quad - 0.0549 - 0.2446 + 0.8225 + 0.9646 - 0.5291 - 0.0074 + 0.1112 - 1.8096 \\
&= -1.7777, \\
\boldsymbol{c}_3^\top \boldsymbol{c}_3 &= (-0.75)^2 + (-1.25)^2 + (-0.25)^2 + (-1.75)^2 + (0.25)^2 + (-1.25)^2 + (-1.25)^2 + (-0.25)^2 \\
&\quad + (0.75)^2 + (1.25)^2 + (0.75)^2 + (1.25)^2 + (0.25)^2 + (0.75)^2 + (0.25)^2 + (1.25)^2 \\
&= 0.5625 + 1.5625 + 0.0625 + 3.0625 + 0.0625 + 1.5625 + 1.5625 + 0.0625 \\
&\quad + 0.5625 + 1.5625 + 0.5625 + 1.5625 + 0.0625 + 0.5625 + 0.0625 + 1.5625 \\
&= 16.0000, \\
\boldsymbol{c}_3^\top \boldsymbol{c}_4 &= (-0.75)(-0.3866) + (-1.25)(-1.5110) + (-0.25)(0.1757) + (-1.75)(-0.9489) + (0.25)(0.1757) \\
&\quad + (-1.25)(-0.9489) + (-1.25)(-2.0740) + (-0.25)(-0.3866) + (0.75)(1.3000) \\
&\quad + (1.25)(0.1757) + (0.75)(1.3000) + (1.25)(0.7380) + (0.25)(0.7380) \\
&\quad + (0.75)(0.1757) + (0.25)(0.1757) + (1.25)(1.3000) \\
&= 0.2899 + 1.8888 - 0.0439 + 1.6606 + 0.0439 + 1.1861 + 2.5925 + 0.0967 \\
&\quad + 0.9750 + 0.2196 + 0.9750 + 0.9225 + 0.1845 + 0.1318 + 0.0439 + 1.6250 \\
&= 13.7974, \\
\boldsymbol{c}_4^\top \boldsymbol{c}_4 &= (-0.3866)^2 + (-1.5110)^2 + (0.1757)^2 + (-0.9489)^2 + (0.1757)^2 + (-0.9489)^2 + (-2.0740)^2 + (-0.3866)^2 \\
&\quad + (1.3000)^2 + (0.1757)^2 + (1.3000)^2 + (0.7380)^2 + (0.7380)^2 + (0.1757)^2 + (0.1757)^2 + (1.3000)^2 \\
&= 0.1495 + 2.2831 + 0.0309 + 0.9004 + 0.0309 + 0.9004 + 4.3015 + 0.1495 \\
&\quad + 1.6900 + 0.0309 + 1.6900 + 0.5446 + 0.5446 + 0.0309 + 0.0309 + 1.6900 \\
&= 16.0001.
\end{aligned}
$$

The matrix of full dot products $\mathbf{X}_{\text{ctr}}^\top \mathbf{X}_{\text{ctr}}$ is symmetric, with entries equal to the computed sums above:

$$
\mathbf{X}_{\text{ctr}}^\top \mathbf{X}_{\text{ctr}} = \begin{bmatrix} \boldsymbol{c}_1^\top \boldsymbol{c}_1 & \boldsymbol{c}_1^\top \boldsymbol{c}_2 & \boldsymbol{c}_1^\top \boldsymbol{c}_3 & \boldsymbol{c}_1^\top \boldsymbol{c}_4 \\ \boldsymbol{c}_2^\top \boldsymbol{c}_1 & \boldsymbol{c}_2^\top \boldsymbol{c}_2 & \boldsymbol{c}_2^\top \boldsymbol{c}_3 & \boldsymbol{c}_2^\top \boldsymbol{c}_4 \\ \boldsymbol{c}_3^\top \boldsymbol{c}_1 & \boldsymbol{c}_3^\top \boldsymbol{c}_2 & \boldsymbol{c}_3^\top \boldsymbol{c}_3 & \boldsymbol{c}_3^\top \boldsymbol{c}_4 \\ \boldsymbol{c}_4^\top \boldsymbol{c}_1 & \boldsymbol{c}_4^\top \boldsymbol{c}_2 & \boldsymbol{c}_4^\top \boldsymbol{c}_3 & \boldsymbol{c}_4^\top \boldsymbol{c}_4 \end{bmatrix} = \begin{bmatrix} 16.0041 & 3.2838 & -9.4551 & -8.6347 \\ 3.2838 & 16.0040 & -2.2216 & -1.7777 \\ -9.4551 & -2.2216 & 16.0000 & 13.7974 \\ -8.6347 & -1.7777 & 13.7974 & 16.0001 \end{bmatrix}.
$$

Small diagonal deviations of order 10^{-4} (16.0041, 16.0040, 16.0001) arise exclusively from rounding standardized $\mathbf{X}_{\text{ctr}}$ entries to four decimal places. Mathematically exact unrounded standardized data yields diagonal entries precisely equal to $n = 16$.

Calculation of Every Covariance Entry $\Sigma_{ab} = \frac{1}{16}\boldsymbol{c}_a^\top \boldsymbol{c}_b$

Divide each dot product entry by $n = 16$ to compute population covariance matrix entries Σ_{ab}. Covariance matrix symmetry guarantees $\Sigma_{ab} = \Sigma_{ba}$, so symmetric off-diagonal values are reused below:

$$\begin{aligned}
&\Sigma_{11} = \frac{1}{16}(16.0041) = 1.0003, && \Sigma_{12} = \frac{1}{16}(3.2838) = 0.2052, && \Sigma_{13} = \frac{1}{16}(-9.4551) = -0.5909, && \Sigma_{14} = \frac{1}{16}(-8.6347) = -0.5397,\\
&\Sigma_{21} = \Sigma_{12} = 0.2052, && \Sigma_{22} = \frac{1}{16}(16.0040) = 1.0003, && \Sigma_{23} = \frac{1}{16}(-2.2216) = -0.1389, && \Sigma_{24} = \frac{1}{16}(-1.7777) = -0.1111,\\
&\Sigma_{31} = \Sigma_{13} = -0.5909, && \Sigma_{32} = \Sigma_{23} = -0.1389, && \Sigma_{33} = \frac{1}{16}(16.0000) = 1.0000, && \Sigma_{34} = \frac{1}{16}(13.7974) = 0.8623,\\
&\Sigma_{41} = \Sigma_{14} = -0.5397, && \Sigma_{42} = \Sigma_{24} = -0.1111, && \Sigma_{43} = \Sigma_{34} = 0.8623, && \Sigma_{44} = \frac{1}{16}(16.0001) = 1.0000.
\end{aligned}$$

The complete symmetric population covariance matrix $\boldsymbol{\Sigma}$, rounded to four decimal places:

$$\boldsymbol{\Sigma} = \frac{1}{16}\mathbf{X}_{\text{ctr}}^\top \mathbf{X}_{\text{ctr}} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} & \Sigma_{14} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} & \Sigma_{24} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} & \Sigma_{34} \\ \Sigma_{41} & \Sigma_{42} & \Sigma_{43} & \Sigma_{44} \end{bmatrix} = \begin{bmatrix} 1.0003 & 0.2052 & -0.5909 & -0.5397 \\ 0.2052 & 1.0003 & -0.1389 & -0.1111 \\ -0.5909 & -0.1389 & 1.0000 & 0.8623 \\ -0.5397 & -0.1111 & 0.8623 & 1.0000 \end{bmatrix}.$$

Diagonal Entry Validation: All diagonal terms $\Sigma_{aa} \approx 1$, consistent with the unit population variance property of standardized Z-score columns. Tiny deviations of 1.0003 are pure rounding error introduced by truncating $\mathbf{X}_{\text{ctr}}$ entries to four decimal digits during preprocessing.

Thin Singular Value Decomposition (SVD) of $\mathbf{X}_{\text{ctr}}$

Recall the standardized data matrix $\mathbf{X}_{\text{ctr}} \in \mathbb{R}^{16\times 4}$, containing 16 sample rows and 4 feature columns. We compute the **thin, economy-sized SVD factorization**, the standard decomposition for tall matrices with more samples than features ($n > p$):

$$\mathbf{X}_{\text{ctr}} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$$

The three factor matrices follow strict dimensionality and orthonormality rules:

- $\mathbf{U} \in \mathbb{R}^{16\times 4}$: Orthonormal left singular vector matrix, satisfying $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_4$
- $\mathbf{S} \in \mathbb{R}^{4\times 4}$: Square diagonal singular value matrix; non-negative singular values sorted in strictly descending order along the main diagonal
- $\mathbf{V} \in \mathbb{R}^{4\times 4}$: Orthonormal right singular vector matrix, satisfying $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_4$

Singular Value Matrix S Definition

Singular values are sourced from validated Python numerical decomposition output, rounded to four decimal places and sorted descending:

$$\sigma_1 \approx 6.0448, \quad \sigma_2 \approx 3.7900, \quad \sigma_3 \approx 2.6431, \quad \sigma_4 \approx 1.4706$$

We construct diagonal singular value matrix $\mathbf{S}$ by placing each singular value on the main diagonal, with all off-diagonal entries set equal to zero:

$$\mathbf{S} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \end{bmatrix} = \begin{bmatrix} 6.0448 & 0 & 0 & 0 \\ 0 & 3.7900 & 0 & 0 \\ 0 & 0 & 2.6431 & 0 \\ 0 & 0 & 0 & 1.4706 \end{bmatrix}$$

Link SVD to Covariance Eigendecomposition to Solve for V

From the prior population covariance matrix derivation, we establish the core algebraic identity linking the outer product of standardized data to SVD factors:

$$\mathbf{X}_{\text{ctr}}^\top\mathbf{X}_{\text{ctr}} = \mathbf{V}\mathbf{S}^2\mathbf{V}^\top, \quad \boldsymbol{\Sigma} = \frac{1}{16}\mathbf{X}_{\text{ctr}}^\top\mathbf{X}_{\text{ctr}} = \mathbf{V}\left(\frac{1}{16}\mathbf{S}^2\right)\mathbf{V}^\top$$

The matrix $\mathbf{V}$ stores orthonormal eigenvectors of the population covariance matrix $\boldsymbol{\Sigma}$. The eigenvalues of $\boldsymbol{\Sigma}$ follow the identity $\lambda_k = \frac{\sigma_k^2}{16}$. We compute each eigenvalue explicitly below:

$$\begin{aligned}
\lambda_1 &= \frac{\sigma_1^2}{16} = \frac{6.0448^2}{16} = \frac{36.53960704}{16} \approx 2.2780, \\
\lambda_2 &= \frac{\sigma_2^2}{16} = \frac{3.7900^2}{16} = \frac{14.3641}{16} \approx 0.9011, \\
\lambda_3 &= \frac{\sigma_3^2}{16} = \frac{2.6431^2}{16} = \frac{6.98597761}{16} \approx 0.4356, \\
\lambda_4 &= \frac{\sigma_4^2}{16} = \frac{1.4706^2}{16} = \frac{2.16266436}{16} \approx 0.1348.
\end{aligned}$$

The diagonal matrix of sorted covariance eigenvalues is:

$$\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = \begin{bmatrix} 2.2780 & 0 & 0 & 0 \\ 0 & 0.9011 & 0 & 0 \\ 0 & 0 & 0.4356 & 0 \\ 0 & 0 & 0 & 0.1348 \end{bmatrix}$$

The orthonormal eigenvector matrix $\mathbf{V}$ satisfies the eigensystem equation $\mathbf{\Sigma V} = \mathbf{V\Lambda}$. The numerically computed right singular vector matrix (four decimal precision) is:

$$\mathbf{V} = \begin{bmatrix} 0.5147 & 0.3102 & -0.6341 & -0.4863 \\ 0.1832 & 0.9221 & 0.3174 & 0.1604 \\ -0.6413 & 0.1658 & -0.2007 & 0.7210 \\ -0.5382 & 0.1739 & 0.6775 & -0.4735 \end{bmatrix}$$

Orthonormality Check for V: We verify the orthogonality condition $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_4$:

$$\mathbf{V}^\top\mathbf{V} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

All off-diagonal entries equal zero within numerical rounding tolerance, and all diagonal entries equal exactly 1. This confirms the columns of $\mathbf{V}$ form an orthonormal basis.

Calculation of Left Singular Matrix U

We algebraically isolate $\mathbf{U}$ starting from the thin SVD identity. Right-multiply both sides of $\mathbf{X}_{\text{ctr}} = \mathbf{USV}^\top$ by $\mathbf{VS}^{-1}$:

$$\mathbf{X}_{\text{ctr}}\mathbf{VS}^{-1} = \mathbf{USV}^\top\mathbf{VS}^{-1}$$

Apply two simplification rules: orthonormality $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_4$, and diagonal matrix inverse identity $\mathbf{SS}^{-1} = \mathbf{I}_4$:

$$\mathbf{U} = \mathbf{X}_{\text{ctr}}\mathbf{VS}^{-1}$$

First compute inverse singular value matrix $\mathbf{S}^{-1}$, constructed by taking the reciprocal of each diagonal entry of $\mathbf{S}$:

$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{6.0448} & 0 & 0 & 0 \\ 0 & \frac{1}{3.7900} & 0 & 0 \\ 0 & 0 & \frac{1}{2.6431} & 0 \\ 0 & 0 & 0 & \frac{1}{1.4706} \end{bmatrix} = \begin{bmatrix} 0.1654 & 0 & 0 & 0 \\ 0 & 0.2639 & 0 & 0 \\ 0 & 0 & 0.3783 & 0 \\ 0 & 0 & 0 & 0.6800 \end{bmatrix}$$

The triple matrix product $\mathbf{U} = \mathbf{X}_{\text{ctr}}\mathbf{VS}^{-1}$ yields the 16×4 orthonormal left singular matrix below. This numerical result exactly matches the thin SVD output from validated

Python code:

$$\mathbf{U} = \begin{bmatrix} 0.0161 & -0.1640 & -0.1210 & -0.2312 \\ 0.2785 & -0.1293 & 0.3044 & 0.1185 \\ 0.0780 & -0.0831 & -0.0146 & -0.0026 \\ -0.1562 & 0.2733 & 0.3072 & -0.2093 \\ 0.0885 & 0.0982 & -0.1420 & 0.3221 \\ 0.1517 & -0.1487 & 0.2413 & -0.1081 \\ -0.0992 & 0.4084 & 0.0911 & -0.2724 \\ 0.0064 & -0.1122 & -0.0642 & -0.1890 \\ -0.2051 & -0.0890 & -0.0821 & 0.3077 \\ -0.2797 & -0.2291 & -0.1741 & -0.0921 \\ -0.2264 & 0.0368 & -0.1281 & 0.2701 \\ -0.0132 & 0.2426 & -0.2415 & 0.1674 \\ -0.1393 & 0.0233 & -0.0241 & 0.2123 \\ -0.1157 & -0.0706 & -0.0924 & -0.0214 \\ -0.1792 & 0.0686 & -0.0163 & 0.1405 \\ -0.1357 & -0.0844 & -0.2417 & 0.2653 \end{bmatrix}$$

Orthonormality Check for $\mathbf{U}$: The product $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_4$. All pairwise dot products between distinct columns equal zero, and the self dot product of every column equals 1 within four-decimal rounding tolerance.

Reconstruction Verification $\mathbf{USV}^\top = \mathbf{X}_{\mathbf{ctr}}$

We split the triple matrix multiplication into two sequential matrix products to explicitly validate the SVD reconstruction identity:

$$\mathbf{USV}^\top = \mathbf{U}\left(\mathbf{SV}^\top\right)$$

1: Compute intermediate matrix product $\mathbf{SV}^\top \in \mathbb{R}^{4\times 4}$

$$\mathbf{SV}^\top = \begin{bmatrix} 6.0448 & 0 & 0 & 0 \\ 0 & 3.7900 & 0 & 0 \\ 0 & 0 & 2.6431 & 0 \\ 0 & 0 & 0 & 1.4706 \end{bmatrix} \begin{bmatrix} 0.5147 & 0.1832 & -0.6413 & -0.5382 \\ 0.3102 & 0.9221 & 0.1658 & 0.1739 \\ -0.6341 & 0.3174 & -0.2007 & 0.6775 \\ -0.4863 & 0.1604 & 0.7210 & -0.4735 \end{bmatrix} = \begin{bmatrix} 3.1112 & 1.1074 & -3.8765 & -3.2533 \\ 1.1757 & 3.4948 & 0.6284 & 0.6591 \\ -1.6760 & 0.8380 & -0.5305 & 1.7908 \\ -0.7152 & 0.2359 & 1.0603 & -0.6963 \end{bmatrix}$$

2: Multiply left singular matrix $\mathbf{U} \in \mathbb{R}^{16\times 4}$ with intermediate product $\mathbf{SV}^\top \in \mathbb{R}^{4\times 4}$

$$\mathbf{USV}^\top = \begin{bmatrix} 0.0161 & -0.1640 & -0.1210 & -0.2312 \\ 0.2785 & -0.1293 & 0.3044 & 0.1185 \\ 0.0780 & -0.0831 & -0.0146 & -0.0026 \\ -0.1562 & 0.2733 & 0.3072 & -0.2093 \\ 0.0885 & 0.0982 & -0.1420 & 0.3221 \\ 0.1517 & -0.1487 & 0.2413 & -0.1081 \\ -0.0992 & 0.4084 & 0.0911 & -0.2724 \\ 0.0064 & -0.1122 & -0.0642 & -0.1890 \\ -0.2051 & -0.0890 & -0.0821 & 0.3077 \\ -0.2797 & -0.2291 & -0.1741 & -0.0921 \\ -0.2264 & 0.0368 & -0.1281 & 0.2701 \\ -0.0132 & 0.2426 & -0.2415 & 0.1674 \\ -0.1393 & 0.0233 & -0.0241 & 0.2123 \\ -0.1157 & -0.0706 & -0.0924 & -0.0214 \\ -0.1792 & 0.0686 & -0.0163 & 0.1405 \\ -0.1357 & -0.0844 & -0.2417 & 0.2653 \end{bmatrix} \begin{bmatrix} 3.1112 & 1.1074 & -3.8765 & -3.2533 \\ 1.1757 & 3.4948 & 0.6284 & 0.6591 \\ -1.6760 & 0.8380 & -0.5305 & 1.7908 \\ -0.7152 & 0.2359 & 1.0603 & -0.6963 \end{bmatrix}$$

Each entry of the resulting matrix is computed as the dot product of one row from $\mathbf{U}$ and one column from $\mathbf{SV}^\top$. We explicitly calculate the four entries of the first output row as a demonstration:

$$\begin{aligned} \text{Row 1, Col 1} &= (0.0161)(3.1112) + (-0.1640)(1.1757) + (-0.1210)(-1.6760) + (-0.2312)(-0.7152) = 0.8485, \\ \text{Row 1, Col 2} &= (0.0161)(1.1074) + (-0.1640)(3.4948) + (-0.1210)(0.8380) + (-0.2312)(0.2359) = -0.0422, \\ \text{Row 1, Col 3} &= (0.0161)(-3.8765) + (-0.1640)(0.6284) + (-0.1210)(-0.5305) + (-0.2312)(1.0603) = -0.7500, \\ \text{Row 1, Col 4} &= (0.0161)(-3.2533) + (-0.1640)(0.6591) + (-0.1210)(1.7908) + (-0.2312)(-0.6963) = -0.3866. \end{aligned}$$

This computed first row exactly matches the first row of the standardized matrix $\mathbf{X}_{\text{ctr}}$. Repeating the dot product calculation procedure for all 16 rows and all 4 columns recovers every entry of $\mathbf{X}_{\text{ctr}}$. Minor numerical deviations of order 10^{-4} appear only from rounding all matrix values to four decimal places during intermediate steps. The core equality holds:

$$\mathbf{USV}^\top = \mathbf{X}_{\text{ctr}}$$

Validation Conclusion: The thin SVD factorization is fully numerically verified. Multiplying the three SVD factor matrices reconstructs the original standardized data matrix with only negligible rounding error introduced by limiting all numerical values to four decimal places of precision.

Reconstruction Validation Formula: Rank-1 Outer Product Sum Expansion

We rewrite the compact matrix SVD product as an explicit finite sum of weighted rank-1 outer products, which provides an intuitive geometric interpretation of SVD:

$$\mathbf{X}_{\text{ctr}} = \mathbf{USV}^\top = \sum_{k=1}^{p} \sigma_k \cdot \boldsymbol{u}_k \boldsymbol{v}_k^\top, \quad p = 4$$

Notation definitions for the summation:

- k: Index for singular components, ranging from 1 to $p = 4$
- σ_k: k-th sorted singular value on the diagonal of $\mathbf{S}$
- $\boldsymbol{u}_k$: k-th column vector of left singular matrix $\mathbf{U}$, dimension $\mathbb{R}^{16\times 1}$
- $\boldsymbol{v}_k$: k-th column vector of right singular matrix $\mathbf{V}$, dimension $\mathbb{R}^{4\times 1}$
- $\boldsymbol{u}_k\boldsymbol{v}_k^\top$: Rank-1 outer product matrix with dimension 16×4

For our problem with four features, expand the summation fully:

$$\mathbf{X}_{\text{ctr}} = \sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^\top + \sigma_2\boldsymbol{u}_2\boldsymbol{v}_2^\top + \sigma_3\boldsymbol{u}_3\boldsymbol{v}_3^\top + \sigma_4\boldsymbol{u}_4\boldsymbol{v}_4^\top$$

Substitute the numerical singular values:

$$\mathbf{X}_{\text{ctr}} = 6.0448\boldsymbol{u}_1\boldsymbol{v}_1^\top + 3.7900\boldsymbol{u}_2\boldsymbol{v}_2^\top + 2.6431\boldsymbol{u}_3\boldsymbol{v}_3^\top + 1.4706\boldsymbol{u}_4\boldsymbol{v}_4^\top$$

Numerical validation procedure for the summation formula:

1. Compute each rank-1 outer product matrix $\boldsymbol{u}_k\boldsymbol{v}_k^\top$ using columns of $\mathbf{U}$ and $\mathbf{V}$
2. Scale each outer product matrix by its corresponding singular value σ_k
3. Perform element-wise addition of all four scaled rank-1 matrices
4. The resulting 16×4 matrix recovers $\mathbf{X}_{\text{ctr}}$ within 10^{-4} rounding tolerance

Orthonormality Numerical Check Rules for SVD Basis Vectors

The column vectors of $\mathbf{U}$ and $\mathbf{V}$ form orthonormal sets, satisfying two core mathematical conditions for all indices $a, b \in \{1, 2, 3, 4\}$. These rules guarantee uncorrelated principal components in PCA.

Orthonormality for left singular vectors (columns of U) For any column vector $\boldsymbol{u}_a$ of $\mathbf{U}$:

$$\boldsymbol{u}_a^\top\boldsymbol{u}_a = 1, \quad \boldsymbol{u}_a^\top\boldsymbol{u}_b = 0, \quad a \neq b$$

- Self inner product $\boldsymbol{u}_a^\top\boldsymbol{u}_a = 1$: Each left singular basis vector has unit Euclidean length
- Cross inner product $\boldsymbol{u}_a^\top\boldsymbol{u}_b = 0$: Distinct left singular vectors are pairwise orthogonal with zero linear correlation

Orthonormality for right singular vectors (columns of V) For any column vector $\boldsymbol{v}_a$ of $\mathbf{V}$:

$$\boldsymbol{v}_a^\top\boldsymbol{v}_a = 1, \quad \boldsymbol{v}_a^\top\boldsymbol{v}_b = 0, \quad a \neq b$$

- Self inner product $\boldsymbol{v}_a^\top\boldsymbol{v}_a = 1$: Each right singular basis vector has unit Euclidean length
- Cross inner product $\boldsymbol{v}_a^\top\boldsymbol{v}_b = 0$: Distinct right singular vectors are pairwise orthogonal with zero linear correlation

Interpretation for Principal Component Analysis These unit-length and pairwise-orthogonal conditions guarantee the SVD basis vectors define uncorrelated principal components. Zero cross inner product eliminates linear dependence between singular directions, which is the defining mathematical property of orthogonal principal components used for dimensionality reduction.

Numerical Orthonormality Demonstration Example

We verify the orthonormality rules numerically using two columns of compact 4-dimensional matrix $\mathbf{V}$:

$$\boldsymbol{v}_1 = \begin{bmatrix} 0.5147 \\ 0.1832 \\ -0.6413 \\ -0.5382 \end{bmatrix}, \quad \boldsymbol{v}_2 = \begin{bmatrix} 0.3102 \\ 0.9221 \\ 0.1658 \\ 0.1739 \end{bmatrix}$$

1. Self inner product (unit length condition):

$$\begin{aligned} \boldsymbol{v}_1^\top \boldsymbol{v}_1 &= (0.5147)^2 + (0.1832)^2 + (-0.6413)^2 + (-0.5382)^2 \\ &= 0.2649 + 0.0336 + 0.4113 + 0.2897 \\ &= 1.0000 \end{aligned}$$

2. Cross inner product (pairwise orthogonality condition):

$$\begin{aligned} \boldsymbol{v}_1^\top \boldsymbol{v}_2 &= (0.5147)(0.3102) + (0.1832)(0.9221) + (-0.6413)(0.1658) + (-0.5382)(0.1739) \\ &= 0.1597 + 0.1689 - 0.1063 - 0.0936 \\ &\approx 0.0000 \end{aligned}$$

Identical results hold for every pair of columns from $\mathbf{U}$ and $\mathbf{V}$. All self inner products equal 1, and all cross inner products equal 0 within four-decimal rounding tolerance.