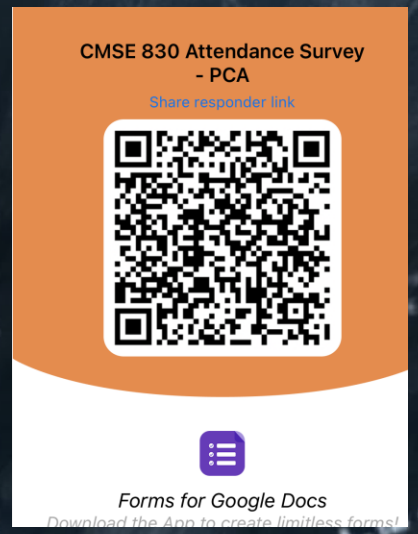


Linear Algebra II

CMSE 830

Review of Multiple Linear Regression

In data science, we usually have many input features (independent variables):

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots$$

In general, every additional feature you add gives you more predictive power.

There are two caveats:

1. The feature x_n is irrelevant. In this case, $w_n=0$; little harm done.
2. If two features are linearly dependent, you need to drop one of them. It adds no new information and it will likely cause mathematical problems.

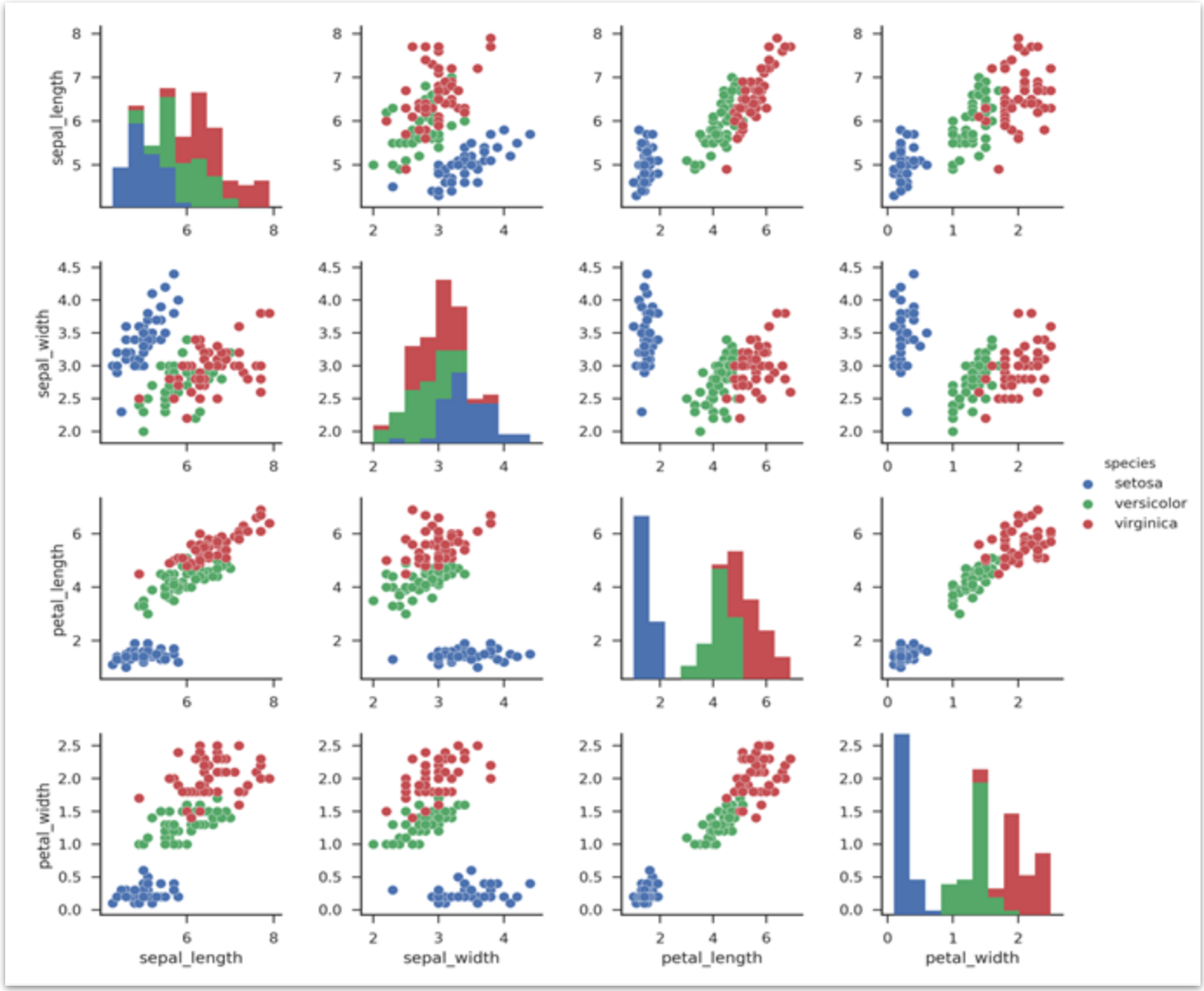
Predicting Petal Width from Features

Suppose we want to predict petal width for versicolor (green).

petal_width = w_0

petal_width = $w_0 + w_1$ petal_length

petal_width = $w_0 + w_1$ petal_length $+ w_2$ sepal_width



Model With Only Bias

$$L = \frac{1}{2}\sum_d (y_d - w_0)^2$$

$$\frac{\partial L}{\partial w_0} = -\sum_d (y_d - w_0) = 0$$

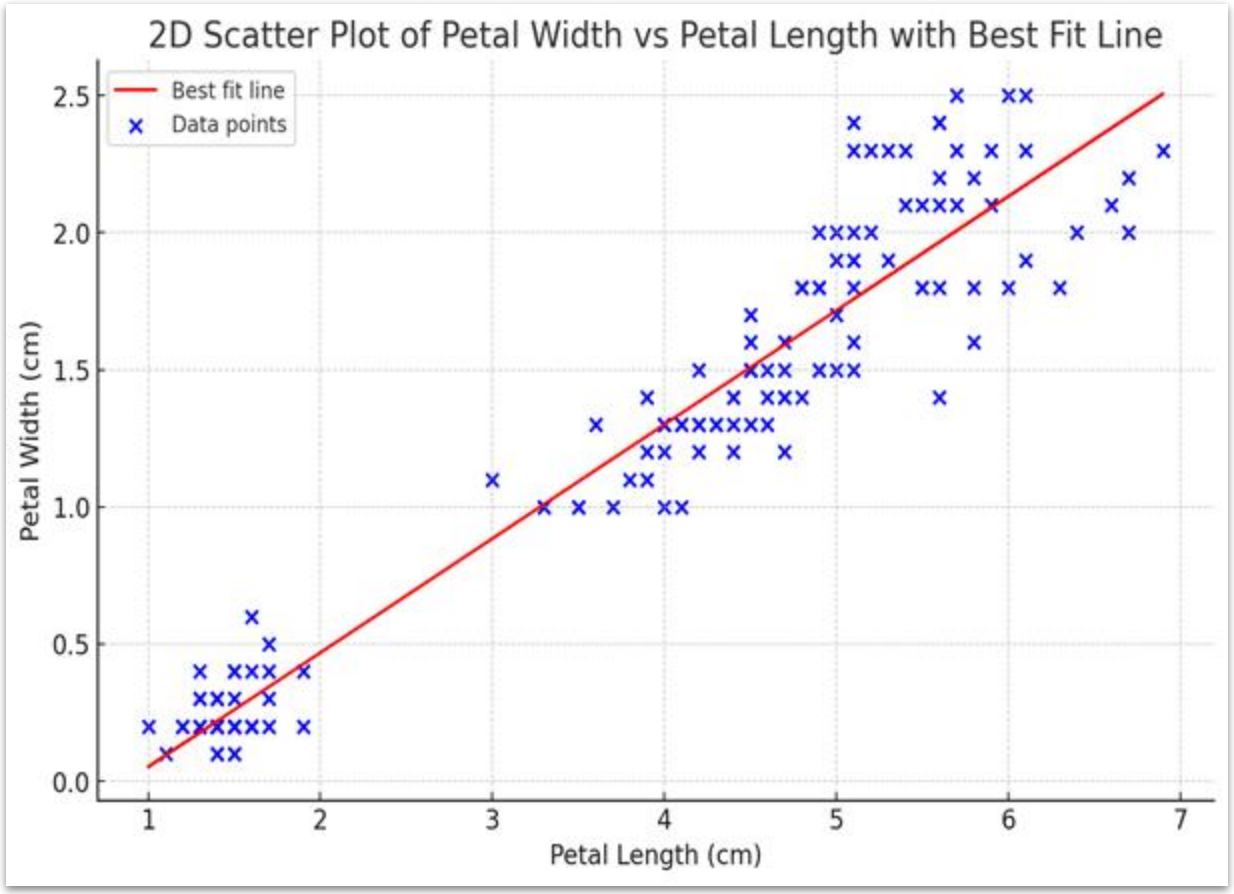
$$\sum_d y_d = \sum_d w_0$$

$$\sum_d y_d = N w_0$$

This shows that the least squares "best" value for the bias is an average of the data.

$$w_0 = \frac{1}{N}\sum_d y_d$$

Multiple Linear Regression with Iris



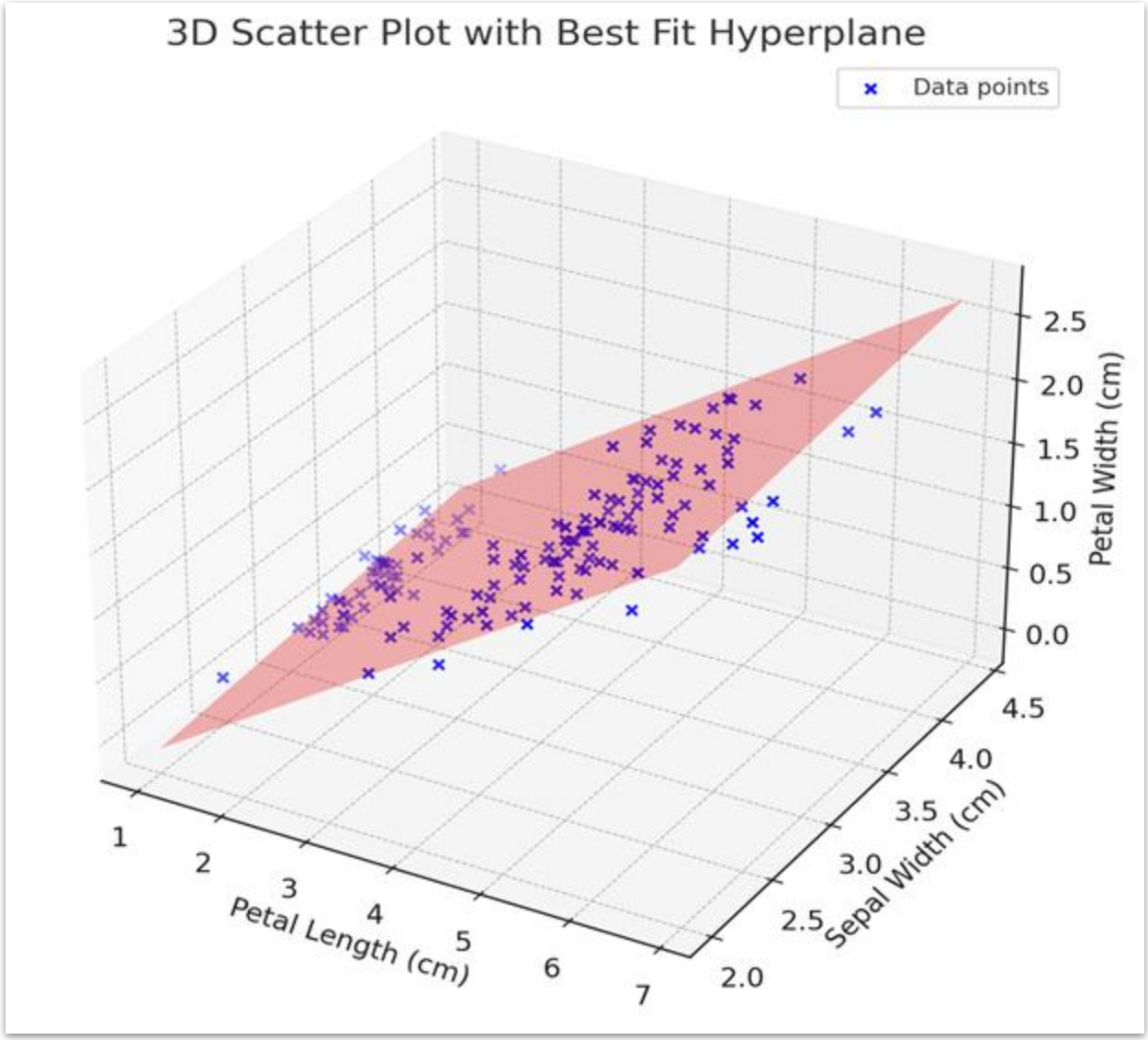


Image Repair with Regression (Interpolation)

Original data

Inverse Reminder

The inverse is defined through:

$$XX^{-1} = I$$

For a 2x2 matrix:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$\det(A) = ad - bc$$

By hand:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} X^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$a = 1$$

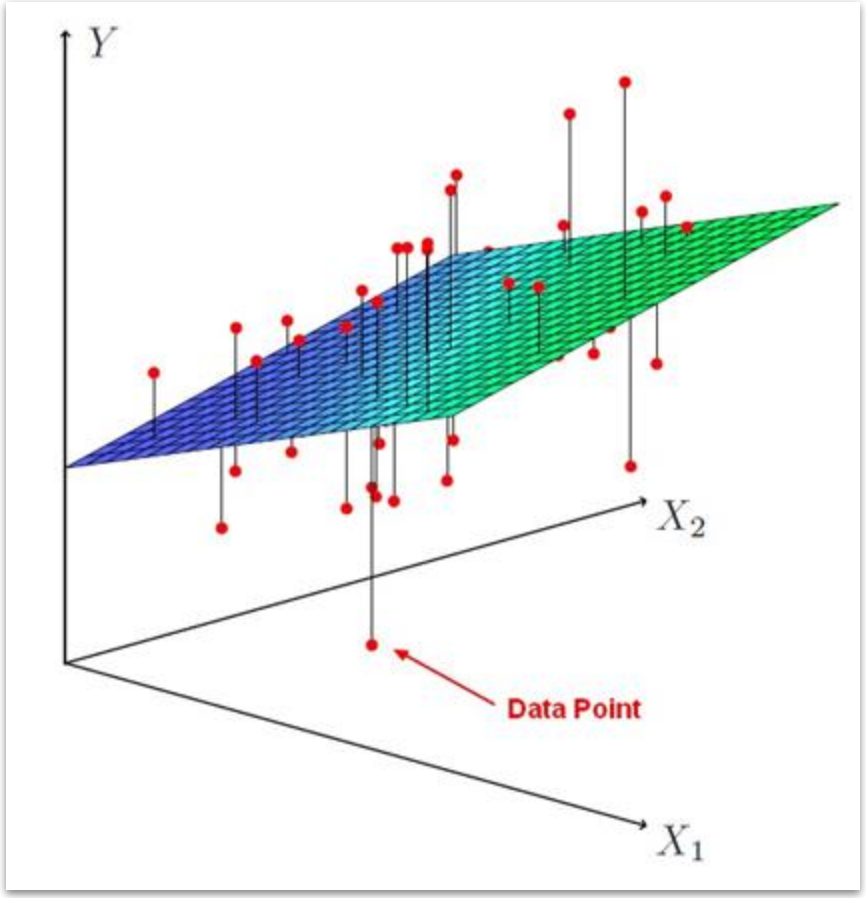
$$b = 0$$

$$a + c = 0$$

$$b + d = 1$$

Note that the determinant must not be zero!

Regression for Non-Square Data Matrices



$$y = w_0 + w_1 x_1 + w_2 x_2,$$

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{N \times 1} = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{21} \\ 1 & x_{12} & x_{22} \\ \vdots & & \\ 1 & x_{1N} & x_{2N} \end{bmatrix}}_{N \times 3} \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}}_{3 \times 1}$$

$$\mathbf{y} = X\mathbf{w}$$

We cannot invert this to find the weights because the data matrix *X* is not square.

Normal Equation

$$\mathbf{y} = X\mathbf{w},$$

$$X^T\mathbf{y} = X^TX\mathbf{w},$$

$$X^T\mathbf{y} = (X^TX)\mathbf{w},$$

$$(X^TX)^{-1}X^T\mathbf{y} = (X^TX)^{-1}(X^TX)\mathbf{w},$$

$$(X^TX)^{-1}X^T\mathbf{y} = I\mathbf{w},$$

$$\boxed{\mathbf{w} = (X^TX)^{-1}X^T\mathbf{y}}$$

Moore-Penrose Pseudoinverse

A generalization of the inverse to non-square matrices is:

$$X^+ = (X^T X)^{-1} X^T$$

In this form, X^+ is referred to as the "*Moore-Penrose inverse*".

Matrix Decompositions

It is often convenient to write a matrix as a product of other matrices.

This might seem like a step backwards, but there are very good reasons for doing this.

Random example:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

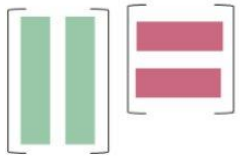
$$A = LU$$

This decomposition is often used to find the inverse and/or determinant of a matrix. You use this when you use `linalg`.

Matrix Decompositions

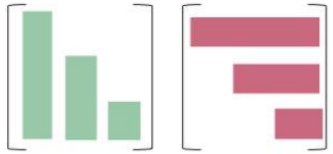
It is very useful to decompose matrices into products of matrices with known and "simple" properties.

$$A = CR$$



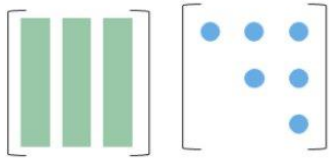
Independent column vectors times row echelon form to show row rank = column rank

$$A = LU$$



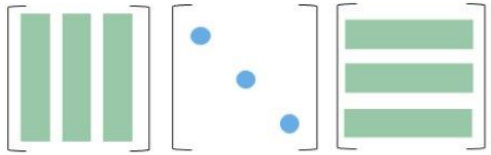
LU decomposition as Gaussian elimination

$$A = QR$$



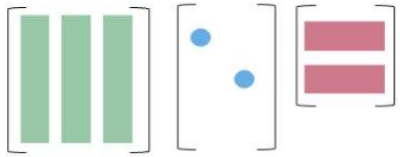
QR decomposition as Gram-Schmidt orthogonalization

$$S = Q\Lambda Q^{\mathrm{T}}$$



Eigenvalue decomposition of a symmetric matrix *S*

$$A = U\Sigma V^{\mathrm{T}}$$



Singular value decomposition of all matrices *A*

Rotation Matrix

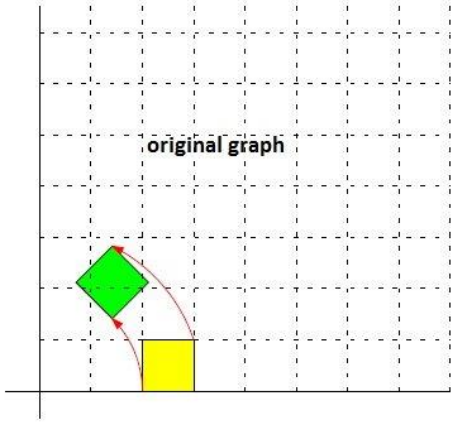
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

To rotate 45° about the origin, we apply the matrix

$$\begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} = \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

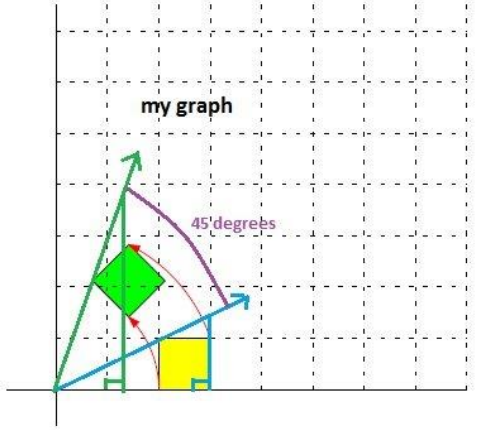
Note: $\frac{\sqrt{2}}{2} = \cos 45^\circ = \sin 45^\circ$, so this is the same as

$$\begin{pmatrix} \cos 45^\circ & -\sin 45^\circ \\ \sin 45^\circ & \cos 45^\circ \end{pmatrix}$$



Counter Clockwise $\begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$

Clockwise $\begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}$



Orthogonal Matrix Q

An orthogonal matrix is a real square matrix with orthonormal vectors as columns and rows.

$$Q^T Q = Q Q^T = I$$

Orthonormal vectors:

$$\mathbf{u}^T \mathbf{v} = 0$$

$$\|\mathbf{u}\| = \|\mathbf{v}\| = 1$$

$$\det(Q) = \pm 1$$

Q preserves distances: $\|Q\mathbf{x}\| = \|\mathbf{x}\|$

$$Q_{\text{rot}} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$Q_{\text{ref}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$Q_{\text{std}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q^T = Q^{-1}$$

Matrix Rank

The rank of a matrix A can be defined in several equivalent ways:

1. Column Rank: Number of linearly independent columns
   - For $A = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & 2 \\ 1 & 3 & 6 \end{bmatrix}$
   - The third column is twice the second column: $\mathbf{c}_3 = 2\mathbf{c}_2$
   - Therefore rank = 2

2. Row Rank: Number of linearly independent rows
   - Same matrix: third row = first row + third row
   - Row rank = Column rank = 2

3. Dimension of the image (range) of A:
   - $\text{rank}(A) = \dim(\text{range}(A))$
   - $\text{range}(A) = \{\mathbf{y} : \mathbf{y} = A\mathbf{x} \text{ for some } \mathbf{x}\}$

Visual interpretation:

- Full rank: Matrix transforms space to same dimension
- Rank deficient: Matrix "squishes" space to lower dimension

Eigenvectors and eigenvalues: Definitions

For a square matrix A, if there exists a non-zero vector $\mathbf{v}$ and scalar λ such that:

$$A\mathbf{v} = \lambda\mathbf{v}$$

Then:

- $\mathbf{v}$ is an eigenvector
- λ is the corresponding eigenvalue

Key Properties:

- Trace = sum of eigenvalues
- Determinant = product of eigenvalues
- If A is symmetric, eigenvalues are real
- If A is orthogonal, $|\lambda| = 1$

Eigen Properties

numpy.linalg.eig

linalg.eig(*a*) [source]

Compute the eigenvalues and right eigenvectors of a square array.

Parameters:

a : *(..., M, M) array*

Matrices for which the eigenvalues and right eigenvectors will be computed

Returns:

A namedtuple with the following attributes:

eigenvalues : *(..., M) array*

The eigenvalues, each repeated according to its multiplicity. The eigenvalues are not necessarily ordered. The resulting array will be of complex type, unless the imaginary part is zero in which case it will be cast to a real type. When *a* is real the resulting eigenvalues will be real (0 imaginary part) or occur in conjugate pairs

eigenvectors : *(..., M, M) array*

The normalized (unit "length") eigenvectors, such that the column `eigenvectors[:,i]` is the eigenvector corresponding to the eigenvalue `eigenvalues[i]`.

When A acts on $\mathbf{v}$:

- Direction of $\mathbf{v}$ is unchanged
- Length is scaled by λ

Example for rotation matrix:

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- Has complex eigenvalues $e^{\pm i\theta}$
- No real vectors are just scaled!

Example for scaling matrix:

$$A = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$$

- Eigenvectors: $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- Eigenvalues: $\lambda_1 = 3, \lambda_2 = 2$

Finding eigenvalues:

1. Solve characteristic equation: $\det(A - \lambda I) = 0$
2. For each λ, solve $(A - \lambda I)\mathbf{v} = \mathbf{0}$

Standard Basis Vectors

$$\hat{e}_i = \begin{bmatrix} 0 \\ \vdots \\ 1_i \\ \vdots \\ 0 \end{bmatrix} \qquad \hat{e}_i^T = [0, \ldots, 1_i, \ldots, 0]$$

The standard basis forms an orthonormal basis.

Standard Basis Vectors and Operations

$$A\hat{e}_i = i\text{th column of A}$$

$$\hat{e}_i^T A = i\text{th row of A}$$

inner product

$$\hat{e}_i^T \hat{e}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\hat{e}_i \hat{e}_j^T = \text{matrix with 1 in } (i,j) \text{ position, 0 elsewhere}$$

outer product

Matrix Construction with Standard Basis Vectors

diagonal matrix

$$D = \sum_{i=1}^{r} d_i \hat{e}_i \hat{e}_i^T$$

identity matrix

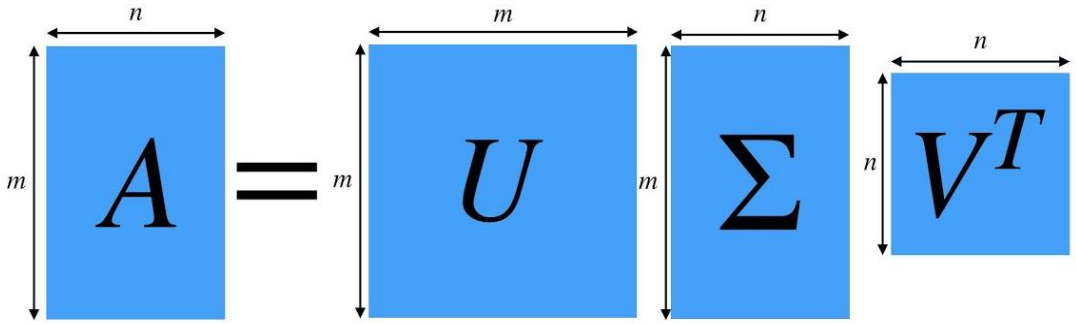
$$I = \sum_{i=1}^{r} \hat{e}_i \hat{e}_i^T$$

SVD Spectral Representation Derivation

$$A = U\Sigma V^T$$

U and V are square and orthogonal.

Σ is not square and is diagonal.



- First, express Σ using basis vectors:

$$\Sigma = \sum_{i=1}^{r} \sigma_i \hat{e}_i \hat{e}_i^T$$

where r is the rank of A

- Substituting into SVD:

$$A = U \left(\sum_{i=1}^{r} \sigma_i \hat{e}_i \hat{e}_i^T \right) V^T$$

- Note that $U\hat{e}_i = \mathbf{u}_i$ (i-th column of U)
- And $V^T \hat{e}_i = \mathbf{v}_i^T$ (i-th row of V^T)

SVD Spectral Representation Derivation

$$
\begin{aligned}
A &= U \left(\sum_{i=1}^{r} \sigma_i \hat{e}_i \hat{e}_i^T \right) V^T \\
&= \sum_{i=1}^{r} \sigma_i (U \hat{e}_i)(\hat{e}_i^T V^T) \\
&= \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T
\end{aligned}
$$

Key observations:

- Each term $\sigma_i \mathbf{u}_i \mathbf{v}_i^T$ is a rank-1 matrix
- The basis vectors $\hat{e}_i$ help us "extract" the corresponding singular vectors
- This representation shows A as a sum of rank-1 matrices

SVD Spectral Representation: Symmetric Matrices

For symmetric matrices where $A = A^T$:

$$\begin{aligned} A &= Q\Lambda Q^T \\ &= Q\left(\sum_{i=1}^{n} \lambda_i \hat{e}_i \hat{e}_i^T\right) Q^T \\ &= \sum_{i=1}^{n} \lambda_i (Q\hat{e}_i)(\hat{e}_i^T Q^T) \\ &= \sum_{i=1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^T \end{aligned}$$

Key connections to SVD:

- When A is symmetric, $U = V = Q$
- Singular values become eigenvalues: $\sigma_i = |\lambda_i|$
- The rank-1 terms are now projections: $\mathbf{q}_i \mathbf{q}_i^T$

Some Applications and Properties

Using these representations, we can easily show:

- Matrix powers:

$$A^k = \sum_{i=1}^{r} \sigma_i^k \mathbf{u}_i \mathbf{v}_i^T$$

- For symmetric matrices:

$$A^k = \sum_{i=1}^{n} \lambda_i^k \mathbf{q}_i \mathbf{q}_i^T$$

- Trace properties:

$$\text{tr}(A) = \sum_{i=1}^{r} \sigma_i = \sum_{i=1}^{n} \lambda_i$$

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$$

- Frobenius norm:

$$\|A\|_F^2 = \sum_{i=1}^{r} \sigma_i^2$$

Regression via SVD

$$\mathbf{y} = K\mathbf{w},$$

K is not square - we can't easily invert this

$$= U\Sigma V^T\mathbf{w},$$

need to define this

$$U^T\mathbf{y} = \Sigma V^T\mathbf{w},$$

$$\Sigma^{-1}U^T\mathbf{y} = V^T\mathbf{w},$$

$$\mathbf{w} = V\Sigma^{-1}U^T\mathbf{y},$$

$$K^+ = V\Sigma^+U^T$$

Σ^+ is the pseudoinverse of Σ, formed by replacing all non-zero entries by their reciprocal and transposing.

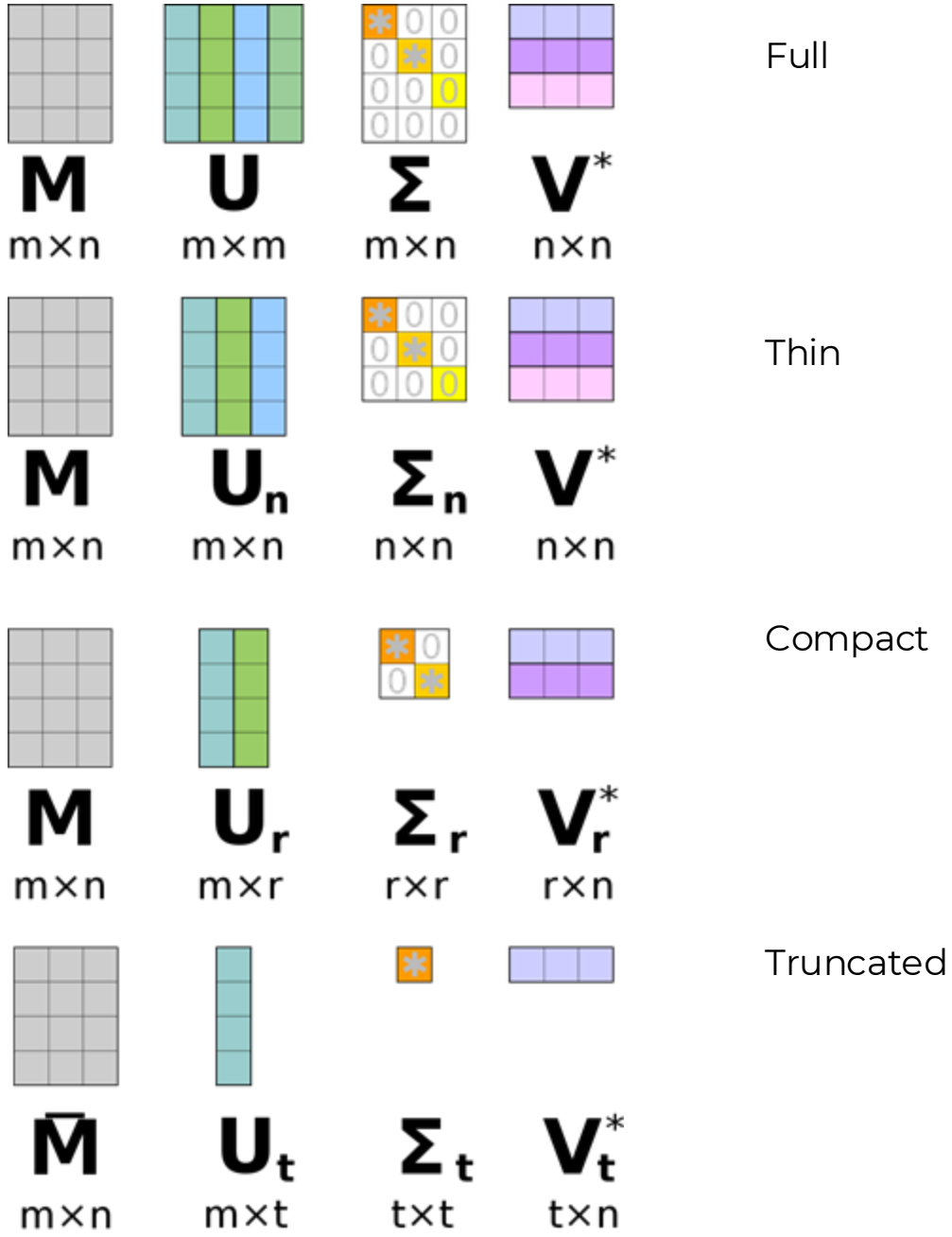
All zero entries remain zero.

Reduced SVDs

Sometimes you will see the SVD in a reduced form.

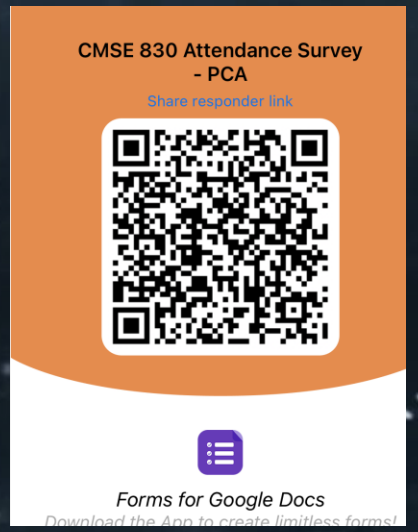
Usually this is for computational reasons. For example, reduced SVDs can save memory.

Often you can tell the library which form you want returned.



Linear Algebra III

- CMSE 830



Statistics Review I

1.1 Means

For total bill (x) and tip (y):

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

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

1.2 Covariance

$$\text{cov}(x, y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

1.3 Covariance Matrix

Unfortunately, the notation Σ for this is the same as used in SVD – they are not the same.

$$\mathbf{\Sigma} = \begin{bmatrix} \text{var}(x) & \text{cov}(x, y) \\ \text{cov}(x, y) & \text{var}(y) \end{bmatrix}$$

1.4 Correlation

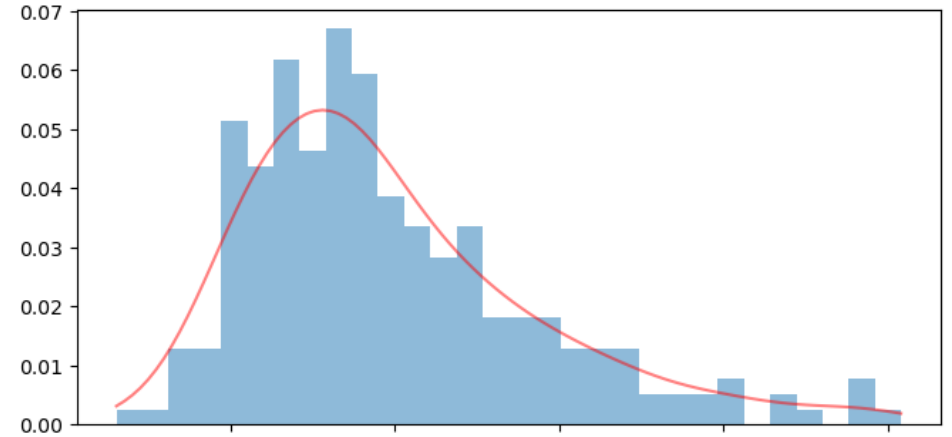
$$\rho_{xy} = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y}$$

where σ_x and σ_y are the standard deviations:

More confusing notation: these are not SVs.

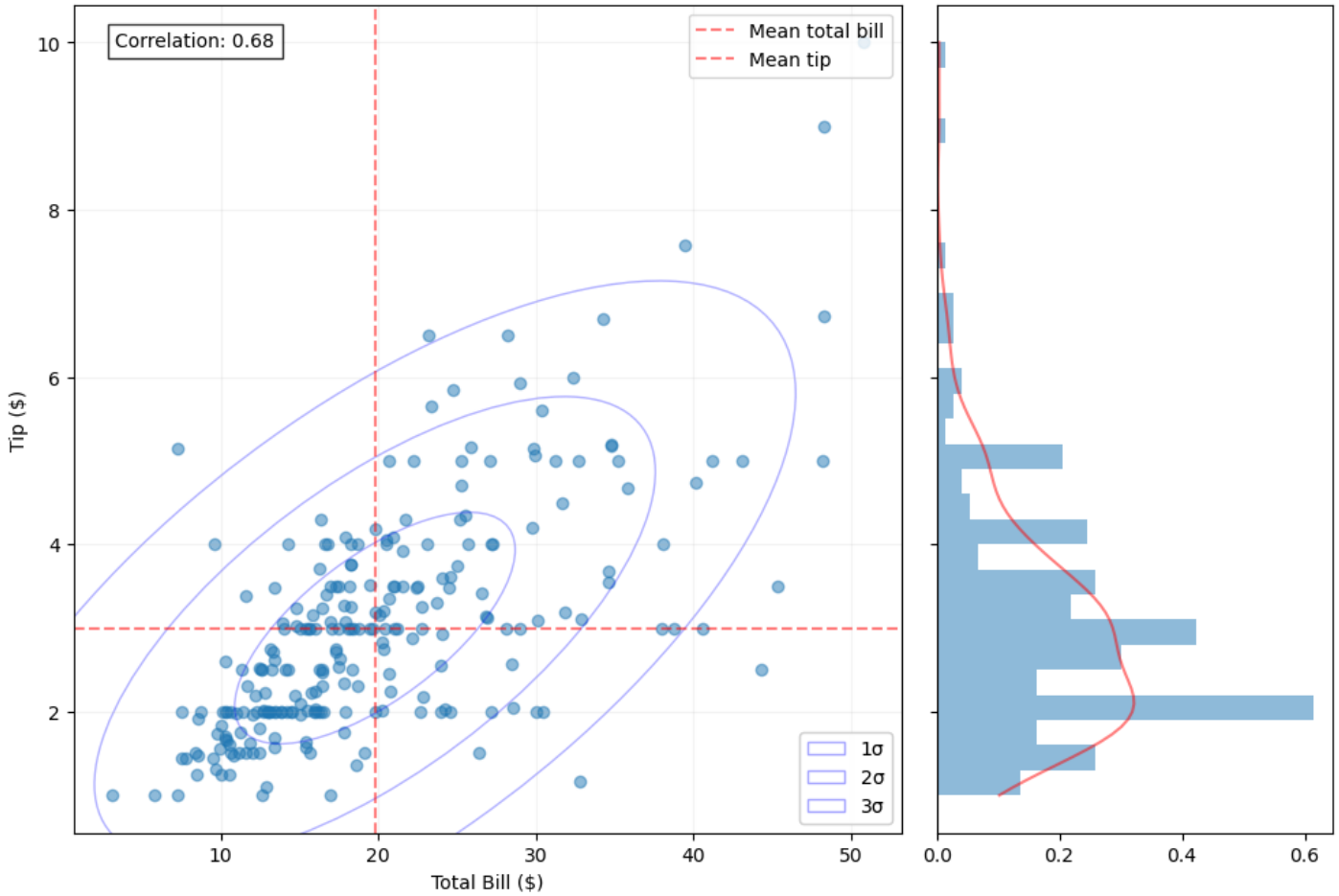
$$\sigma_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\sigma_y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$




```
Summary Statistics:
Mean total bill: $19.79
Mean tip: $3.00
Correlation: 0.68

Covariance matrix:
[[79.25293861  8.32350163]
 [ 8.32350163  1.91445464]]
```



Statistics Review II

2.1 General Form

The equation for the ellipse at k standard deviations is:

$$(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = k^2$$

where:

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$$

3.1 Kernel Density Estimation

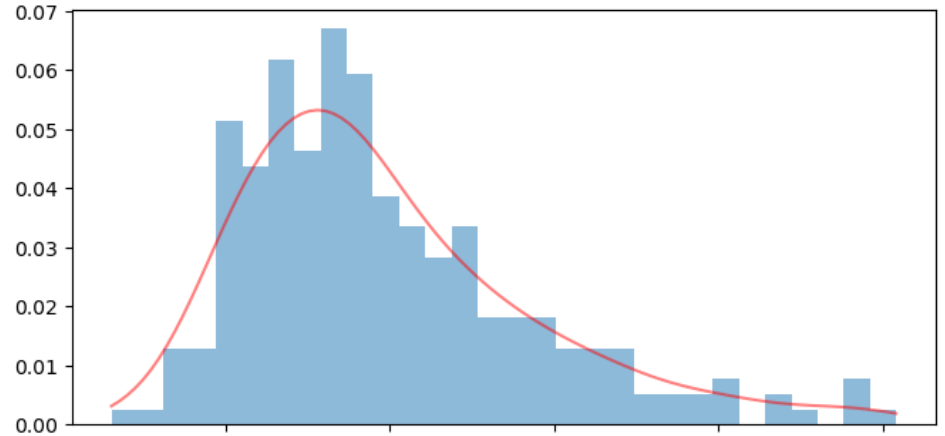
For the marginal distributions:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where K is the kernel function and h is the bandwidth.

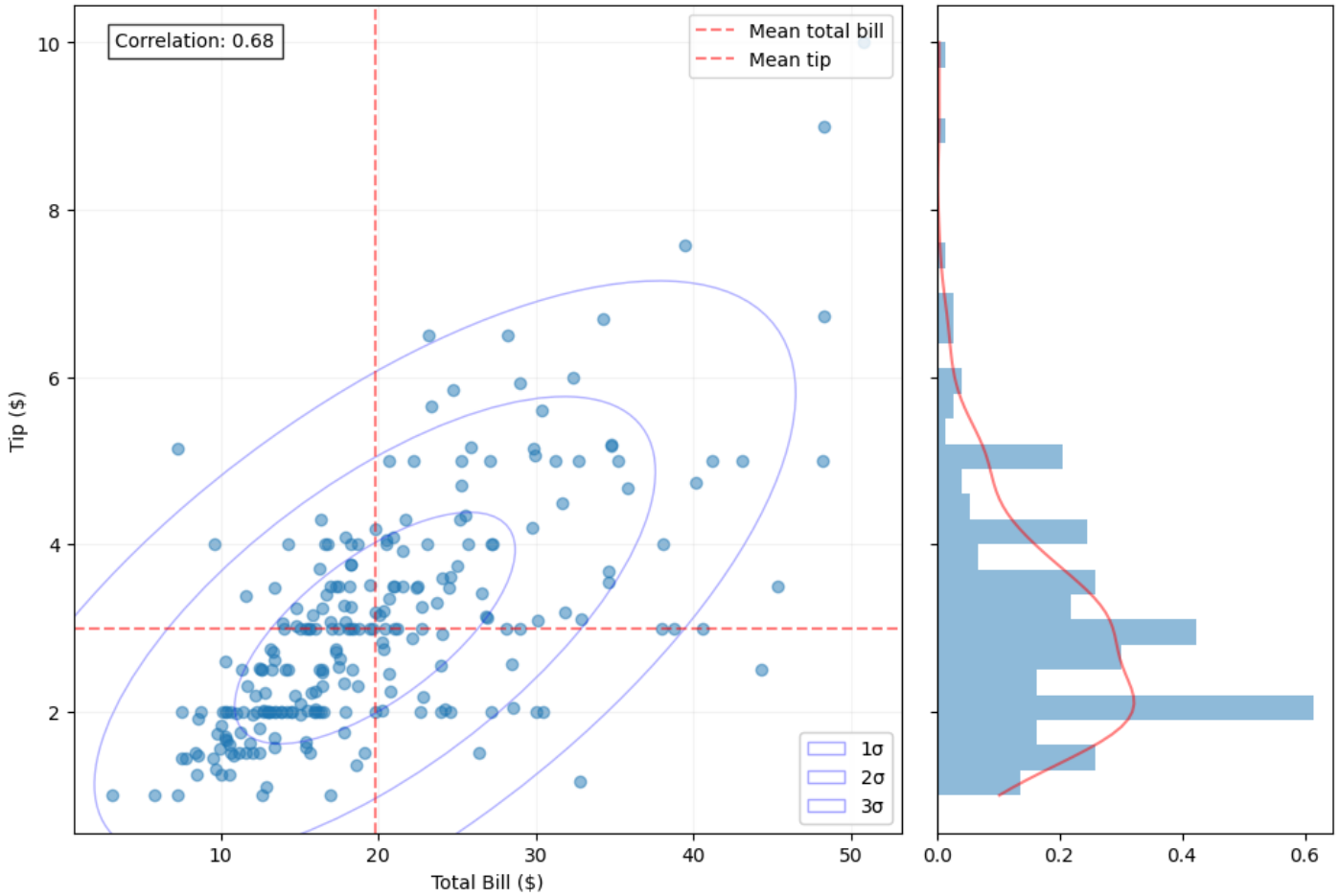
3.2 Gaussian Kernel

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$




```
Summary Statistics:
Mean total bill: $19.79
Mean tip: $3.00
Correlation: 0.68

Covariance matrix:
[[79.25293861  8.32350163]
 [ 8.32350163  1.91445464]]
```



Covariance Matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & \cdots & x_{1p} - \bar{x}_p \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & \cdots & x_{2p} - \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & \cdots & x_{np} - \bar{x}_p \end{bmatrix}$$

Mean-centered data matrix.

We can write $\mathbf{X}$ in terms of its column vectors:

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_p]$$

where each $\mathbf{x}_i$ is a centered column vector:

$$\mathbf{x}_i = \begin{bmatrix} x_{1i} - \bar{x}_i \\ x_{2i} - \bar{x}_i \\ \vdots \\ x_{ni} - \bar{x}_i \end{bmatrix}$$

Now, let's examine $\mathbf{X}^\top \mathbf{X}$:

$$\mathbf{X}^\top \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_p^\top \end{bmatrix} [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_p]$$

This gives us:

$$\mathbf{X}^\top \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \mathbf{x}_1 & \mathbf{x}_1^\top \mathbf{x}_2 & \cdots & \mathbf{x}_1^\top \mathbf{x}_p \\ \mathbf{x}_2^\top \mathbf{x}_1 & \mathbf{x}_2^\top \mathbf{x}_2 & \cdots & \mathbf{x}_2^\top \mathbf{x}_p \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_p^\top \mathbf{x}_1 & \mathbf{x}_p^\top \mathbf{x}_2 & \cdots & \mathbf{x}_p^\top \mathbf{x}_p \end{bmatrix}$$

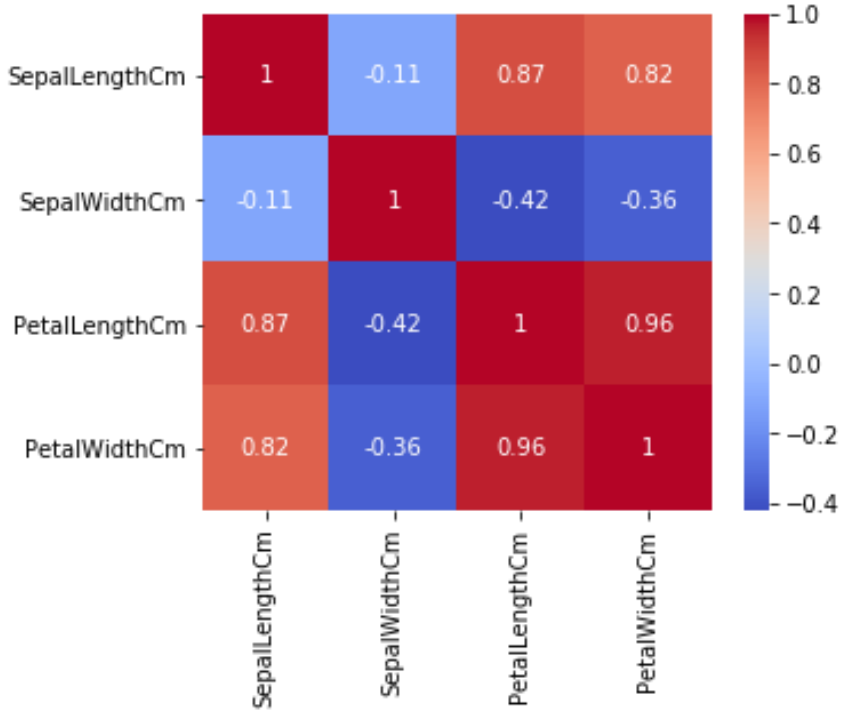
Each element is an inner product:

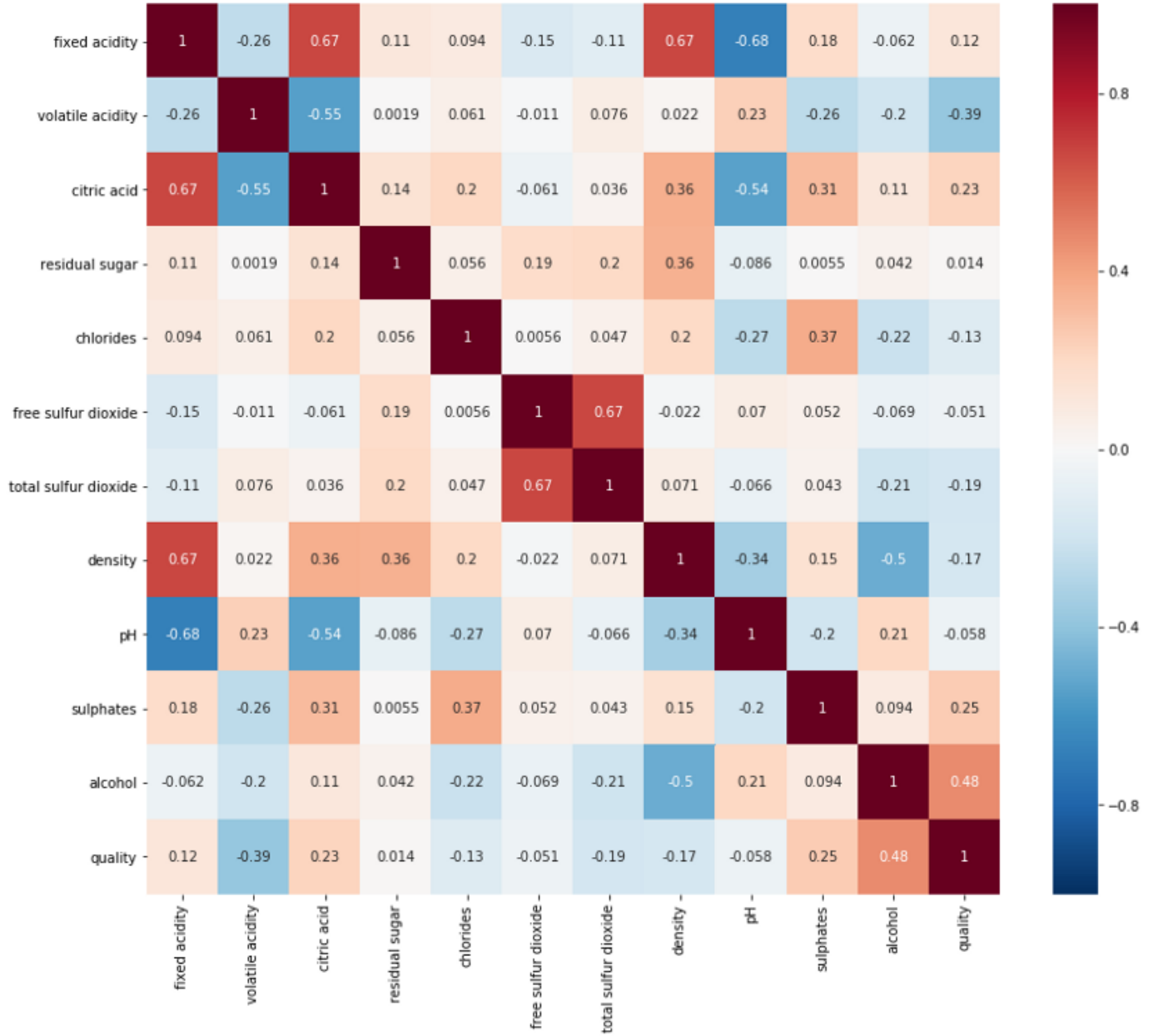
$$(\mathbf{X}^\top \mathbf{X})_{ij} = \mathbf{x}_i^\top \mathbf{x}_j = \sum_{k=1}^{n} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$$

To get the true covariance matrix we need a factor of 1/ (n-1) .

Visualize Correlation

```
correlation = data.corr()
#display(correlation)
plt.figure(figsize=(14, 12))
heatmap = sns.heatmap(correlation, annot=True, linewidths=0, vmin=-1, cmap="RdBu_r")
```





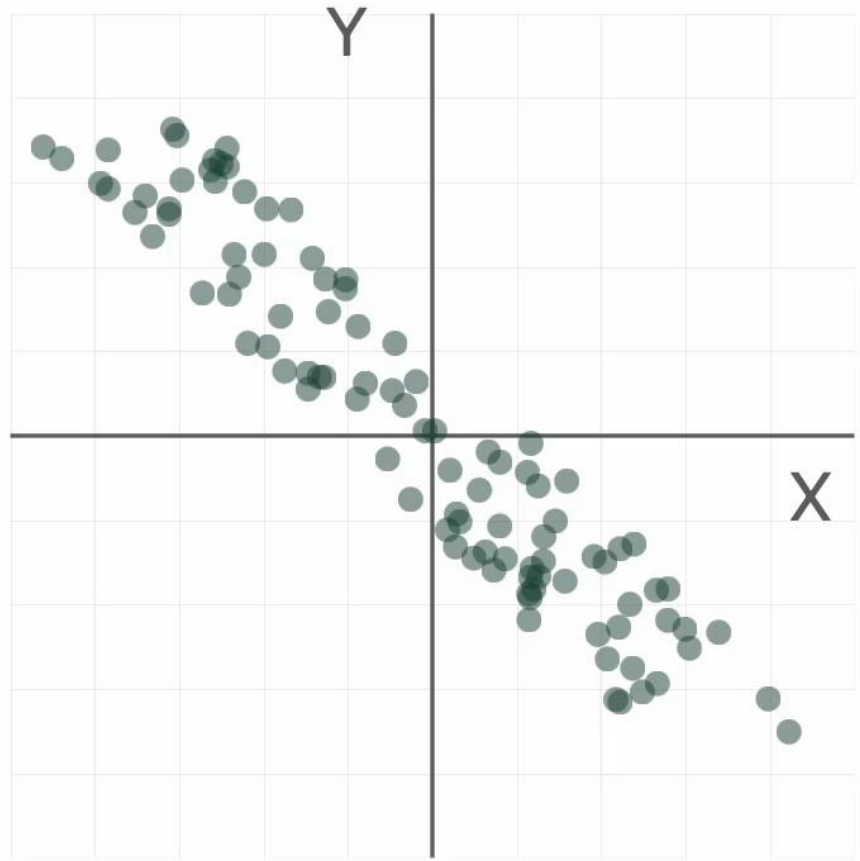
Data Rotation and Covariance Matrix

Play

277°

Reset

Generate New Data



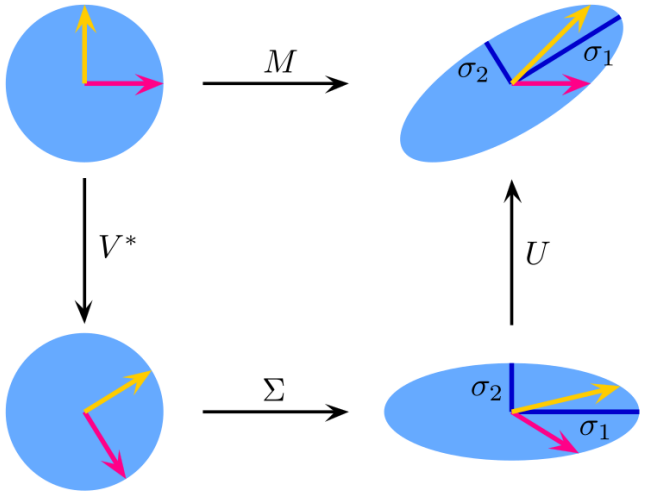
0.287	-0.262
-0.262	0.267

Correlation: -0.947

SVD Review

SVD is important for many reasons:

1. It yields the best low-rank approximation of a matrix.
2. Reveals how matrices transform.
3. Connects many matrix properties (e.g., rank, condition number, matrix norm, range, nullspace, etc.)



$$M = U \cdot \Sigma \cdot V^*$$

For a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, the SVD is:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

where:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m \end{bmatrix} \in \mathbb{R}^{m \times m}$$

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{m \times n}$$

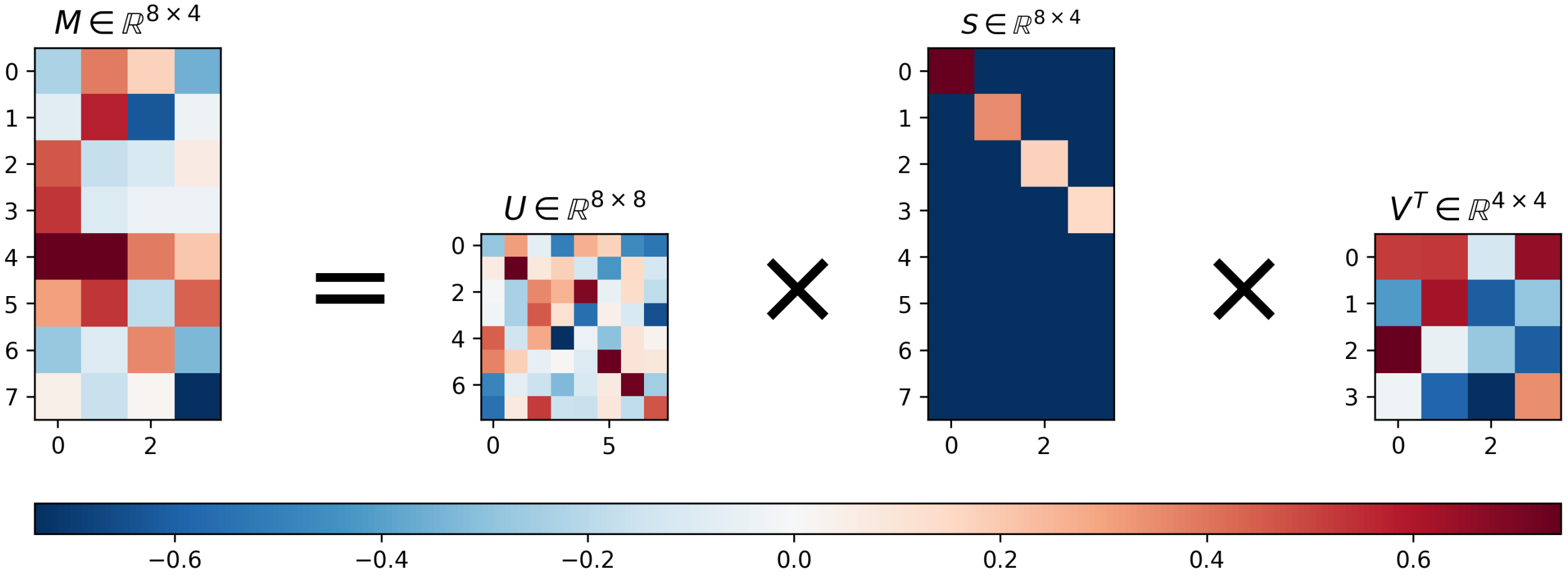
$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix} \in \mathbb{R}^{n \times n}$$

We can write this as a sum of rank-1 matrices:

$$\mathbf{X} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

where $r = \text{rank}(\mathbf{X})$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$

Reducing the Rank with SVD



We can set some of the singular values to zero but keep the shapes of all the matrices fixed.

Recall that the SVs are ordered from largest to smallest.

Spectral Representation: Sum of Rank-1 Matrices

$$A = U\left(\sum_{i=1}^{r} \sigma_i \hat{e}_i \hat{e}_i^T\right) V^T$$

$$= \sum_{i=1}^{r} \sigma_i (U\hat{e}_i)(\hat{e}_i^T V^T)$$

$$= \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

Let $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$ be column vectors:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Their outer product $\mathbf{u}\mathbf{v}^\top$ creates an $m \times n$ matrix:

$$\mathbf{u}\mathbf{v}^\top = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_n \\ u_2 v_1 & u_2 v_2 & \cdots & u_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_m v_1 & u_m v_2 & \cdots & u_m v_n \end{bmatrix}$$

We can set some of the singular values to zero but keep the shapes of all the matrices fixed.

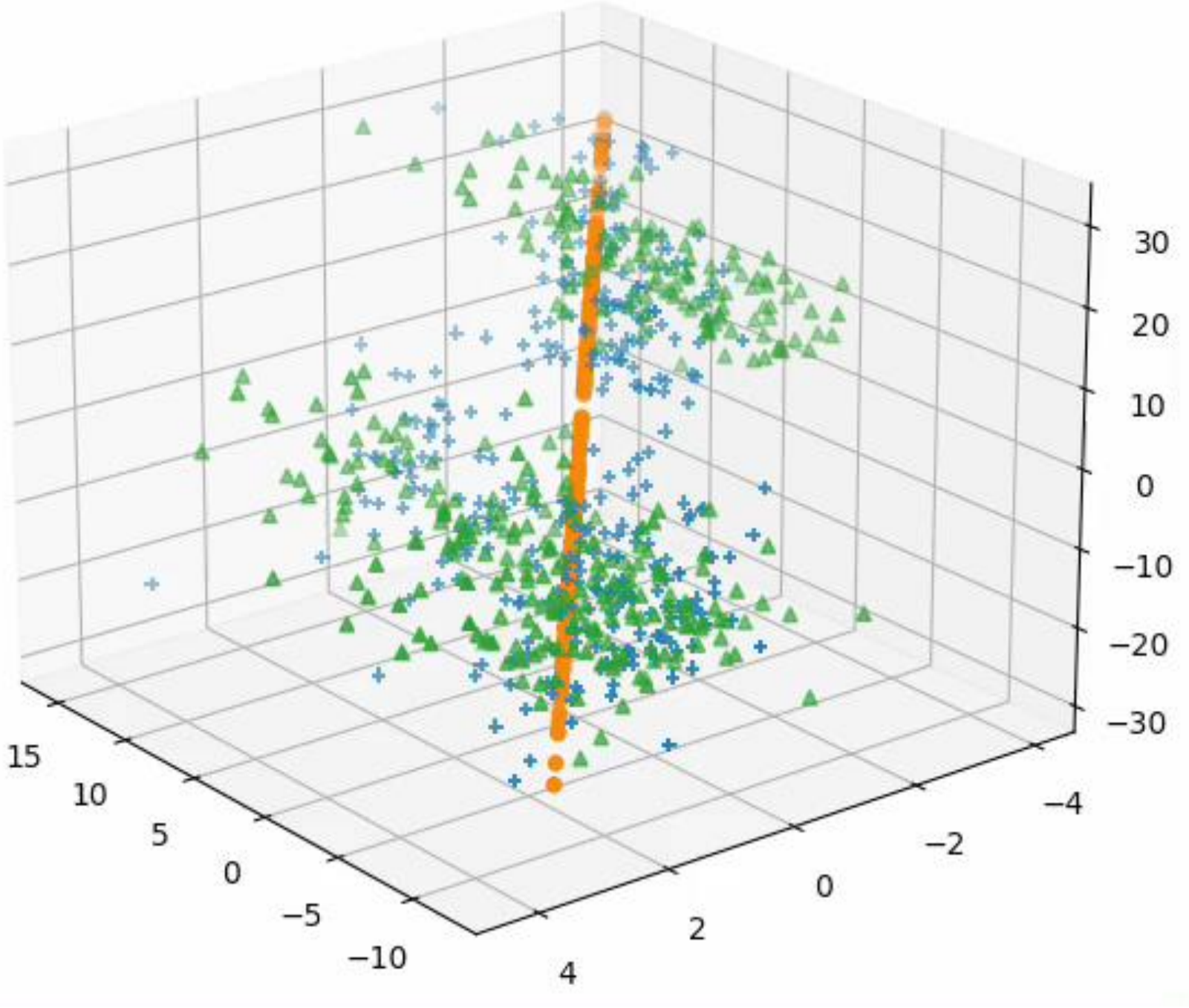
Recall that the SVs are ordered from largest to smallest.

SVD Visualization I

Rank 3 ($\sigma_3, \sigma_2, \sigma_1 > 0$)

Rank 2 ($\sigma_1 = 0$)

Rank 1 ($\sigma_2, \sigma_1 = 0$)



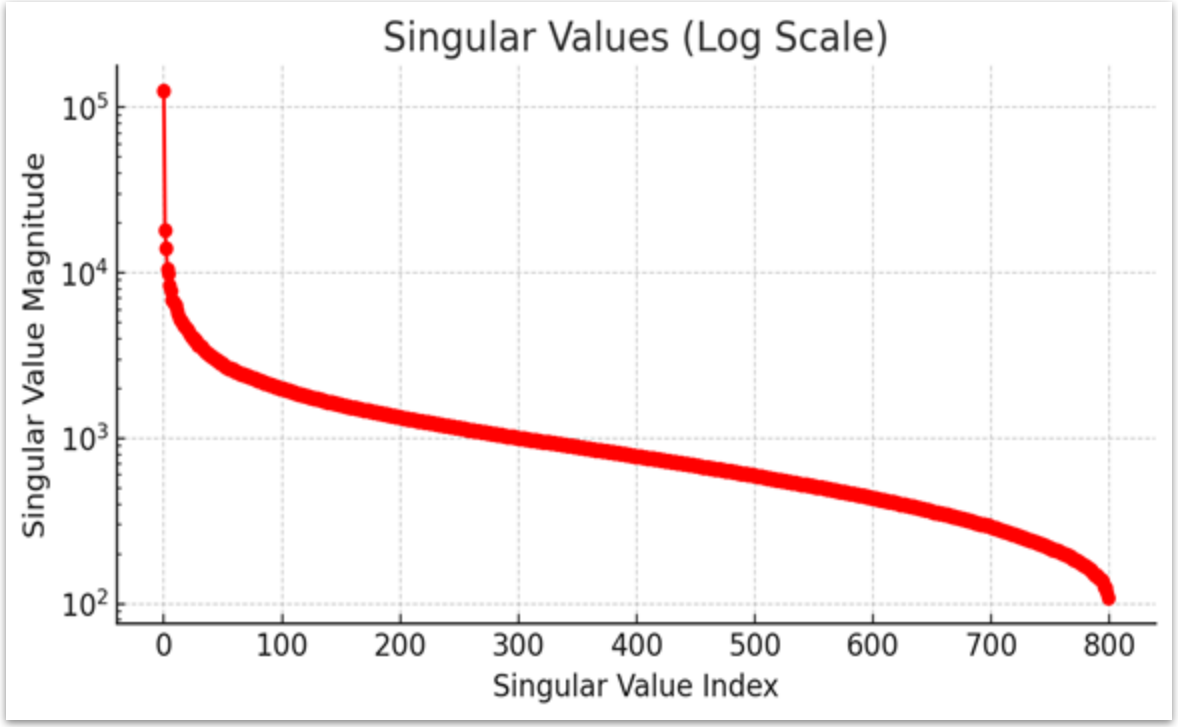
SVD Visualization II

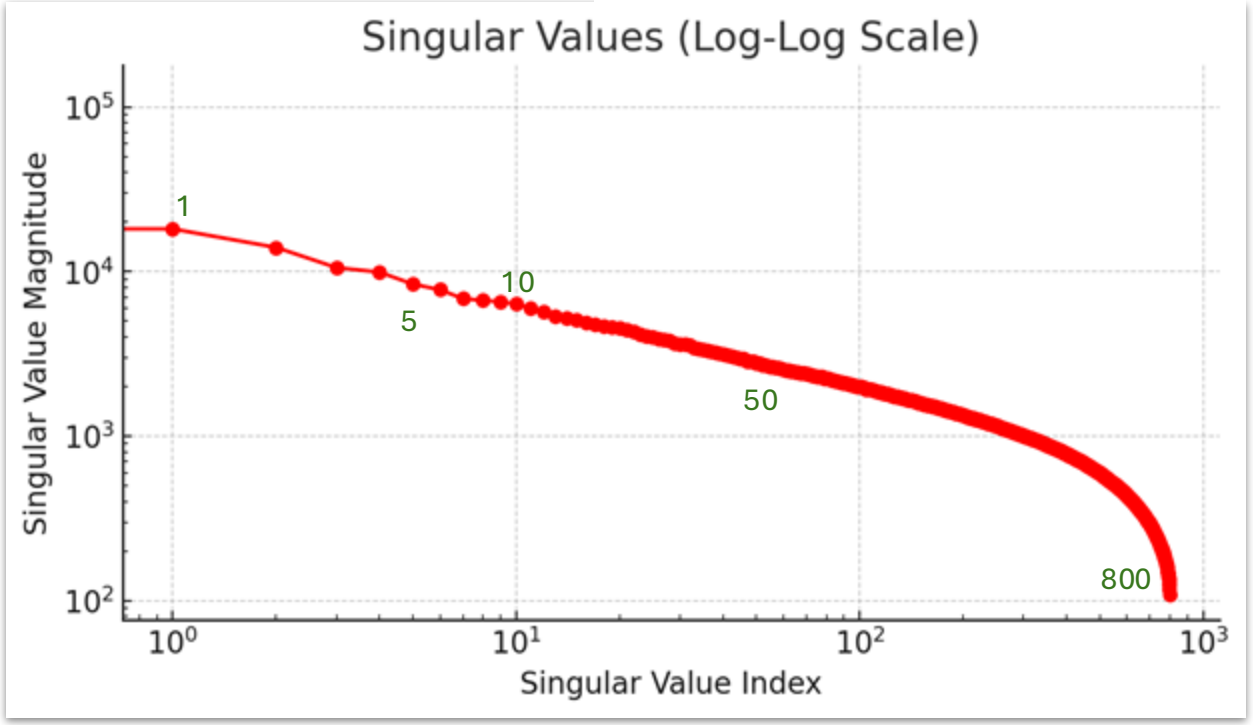
$$A = U\left(\sum_{i=1}^{r}\sigma_i \hat{e}_i \hat{e}_i^T\right)V^T$$

$$= \sum_{i=1}^{r}\sigma_i (U\hat{e}_i)(\hat{e}_i^T V^T)$$

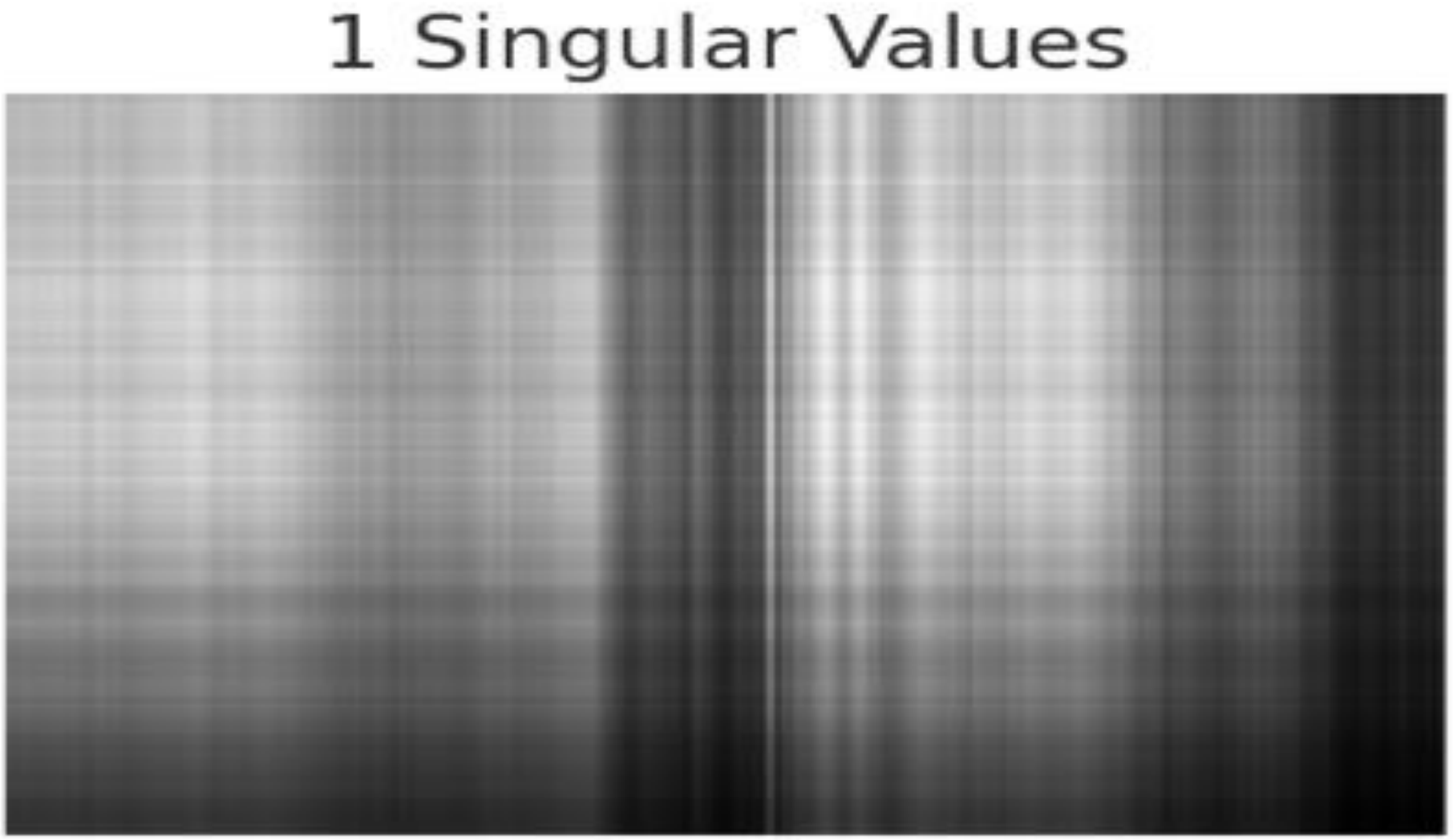
$$= \sum_{i=1}^{r}\sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

Suppose we have an image stored in a data matrix X. We want to compress this image using SVD; that is, we want to expand the image in outer products.

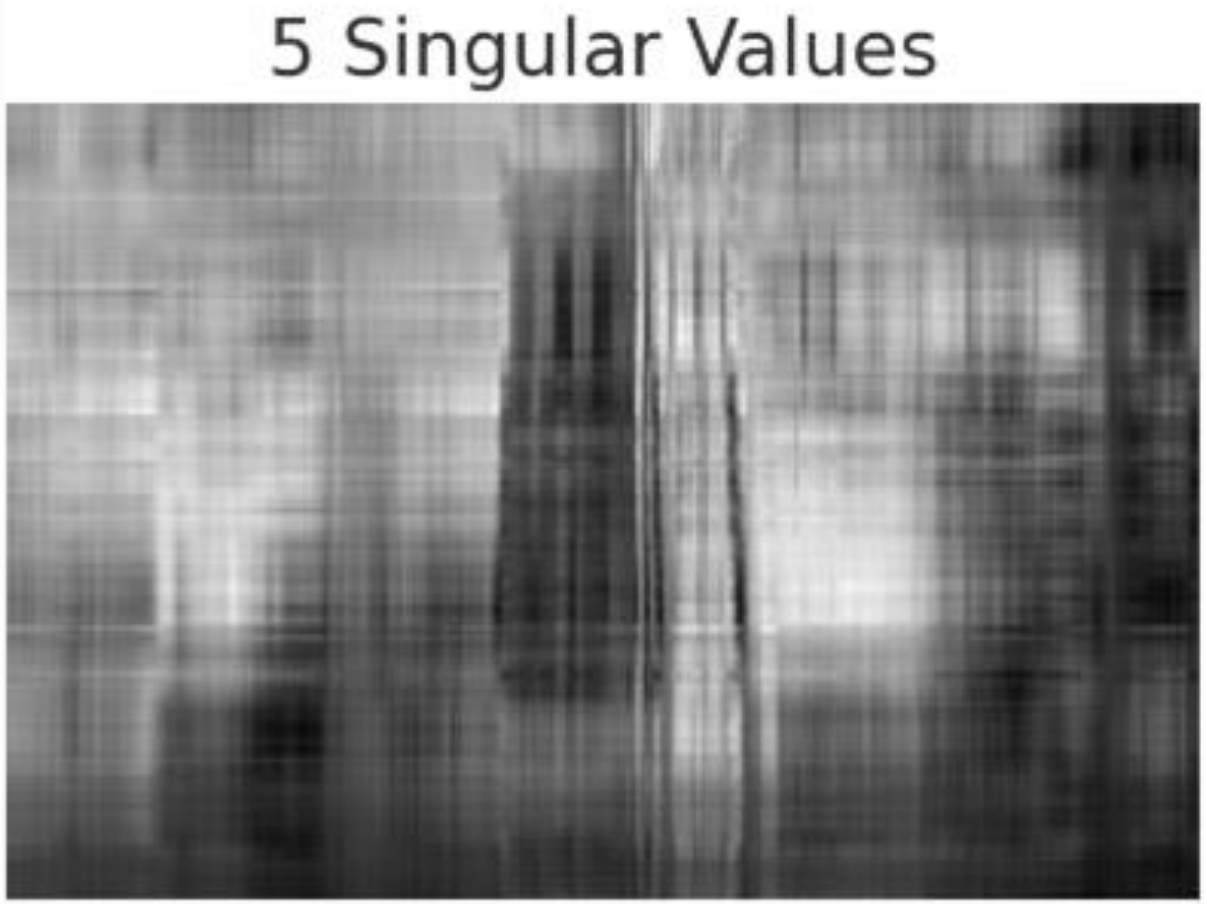


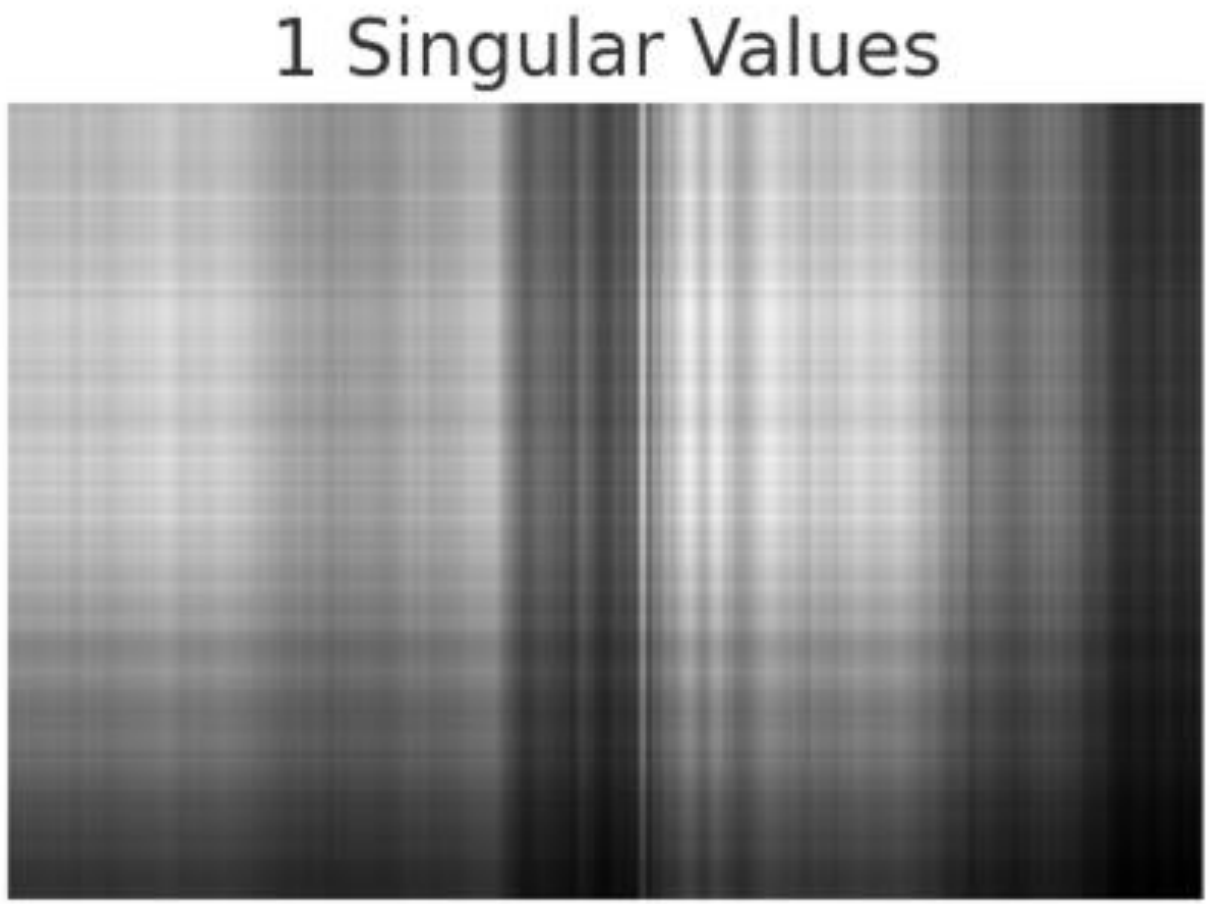


SVD Visualization II: One Singular Value

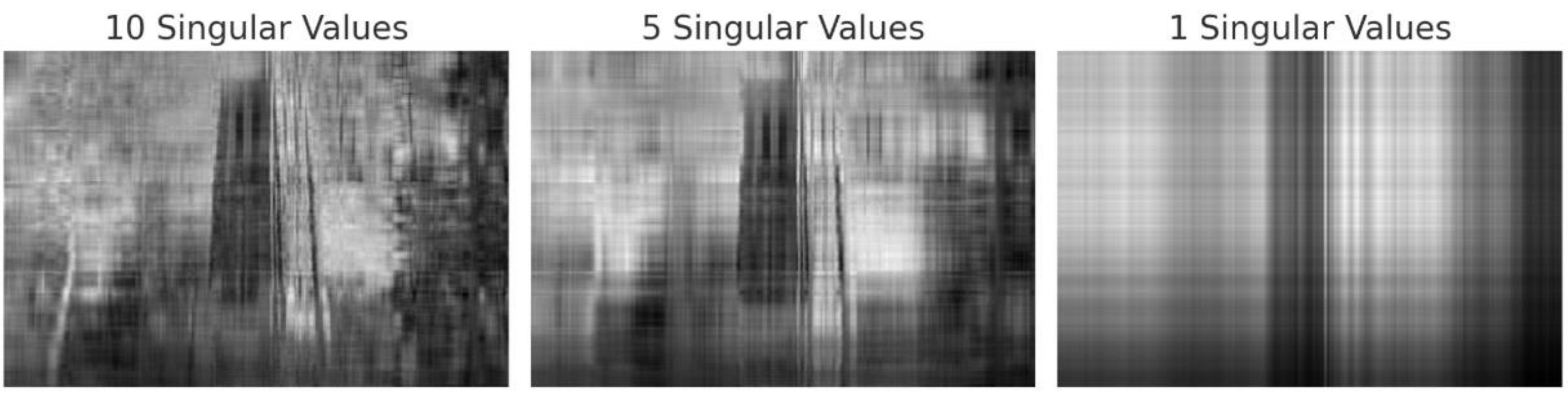


SVD Visualization II: Five Singular Values





SVD Visualization II: Ten Singular Values

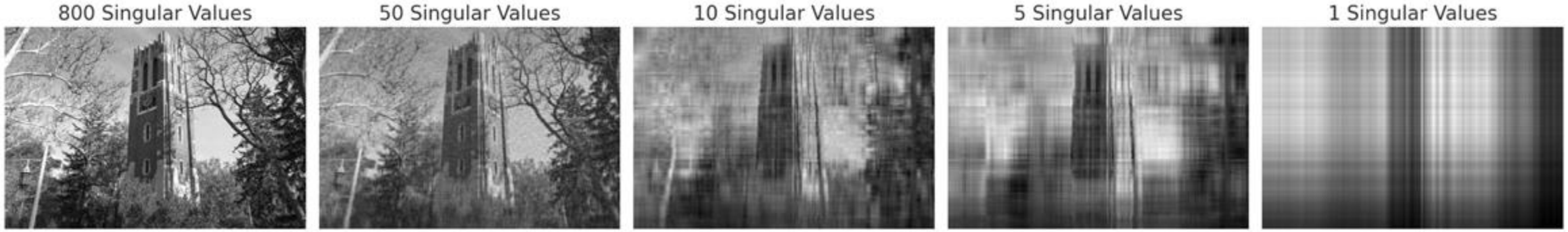


SVD Visualization II: 50 and 800 and Singular Values

800 Singular Values

50 Singular Values

SVD Visualization II: Summary



Correlation Matrix and its SVD

For centered data matrix X (each column has mean zero):

- Correlation matrix $C = X^T X$:

$$C = X^T X = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T \Sigma V^T$$

- Key observations:
  - C is symmetric: $C = C^T$
  - Columns of V are eigenvectors of C
  - Eigenvalues of C are squares of singular values: $\lambda_i = \sigma_i^2$
- For any column $\mathbf{v}_i$ of V:

$$C\mathbf{v}_i = \sigma_i^2 \mathbf{v}_i$$

Variance Explained

- Total variance in data:

$$\text{Total Variance} = \text{tr}(C) = \sum_{i=1}^{r} \sigma_i^2$$

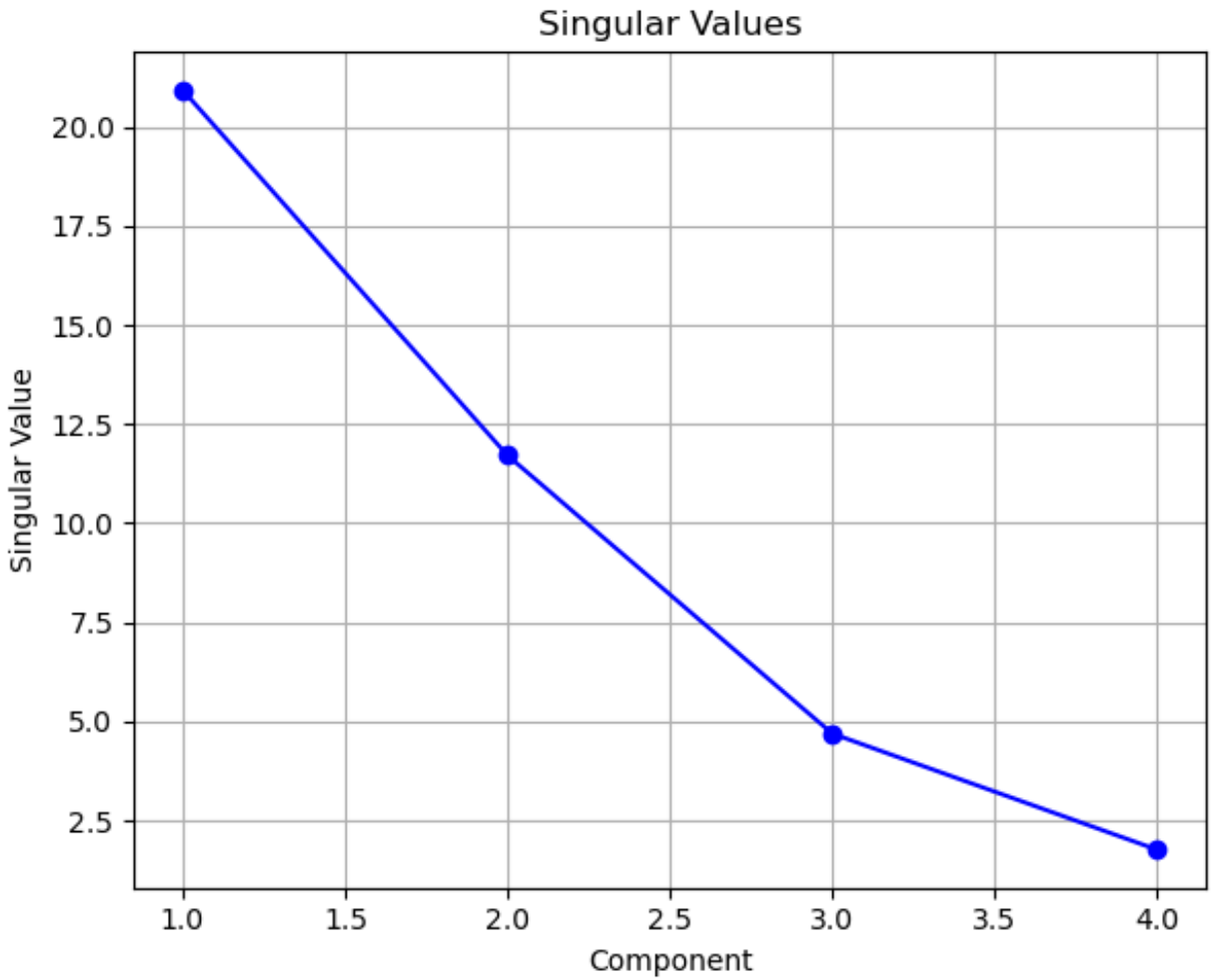
Trace means sum the diagonals.

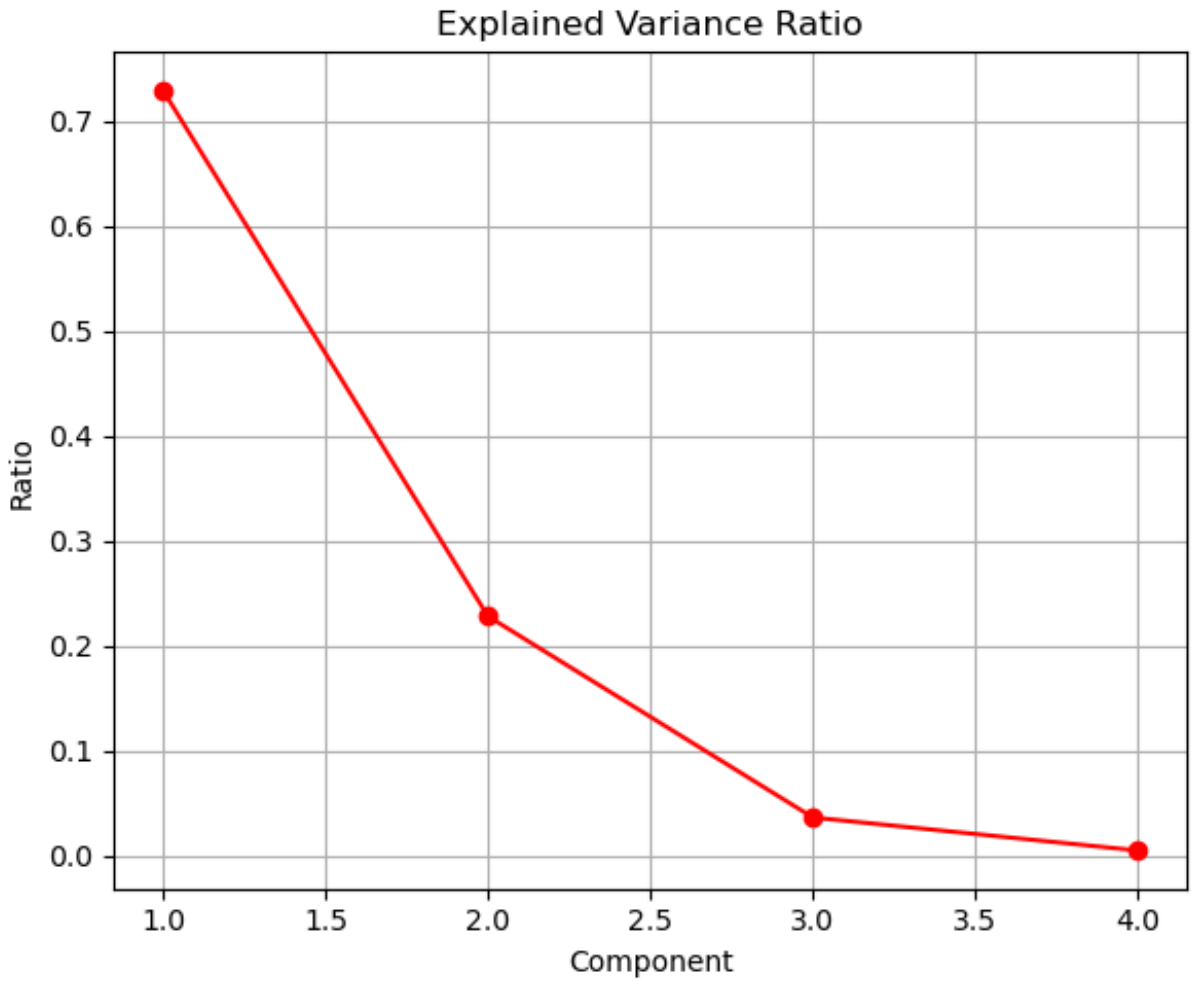
- Proportion of variance explained by kth component:

$$\text{Proportion}_k = \frac{\sigma_k^2}{\sum_{i=1}^{r} \sigma_i^2}$$

- Cumulative variance explained:

$$\text{Cumulative}_k = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2}$$


Singular Values
Singular Value
Component
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Explained Variance Ratio
Ratio
Component
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Principal Component Analysis (PCA): Dimensionality Reduction

Steps for dimensionality reduction:

1. Center the data matrix X
2. Compute SVD: $X = U\Sigma V^T$
3. Choose $k < r$ components based on desired variance explained
4. Project data onto first k principal components:

$$X_k = X(V_k) = U_k\Sigma_k$$

where V_k contains first k columns of V

By dropping columns in V, those associated with the smallest SVs, we obtain a new data matrix with fewer columns.

A potential "problem" is that the features in this new data matrix are less clear.

Key properties:

- X_k is optimal rank-k approximation:

$$X_k = _{\text{rank}(B)=k} \|X - B\|_F$$

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$$

- New coordinates are uncorrelated
- Maximum variance preserved in each direction

Scree Plots and the Elbow Method

Formal Definition of Scree Plot

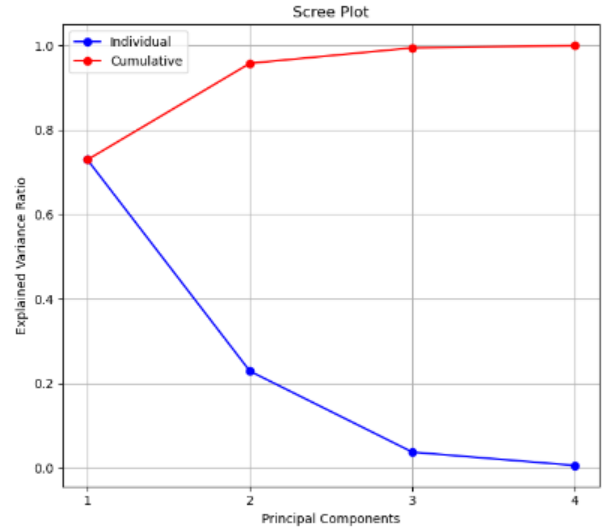
A scree plot visualizes the ordered singular values or their derived quantities:

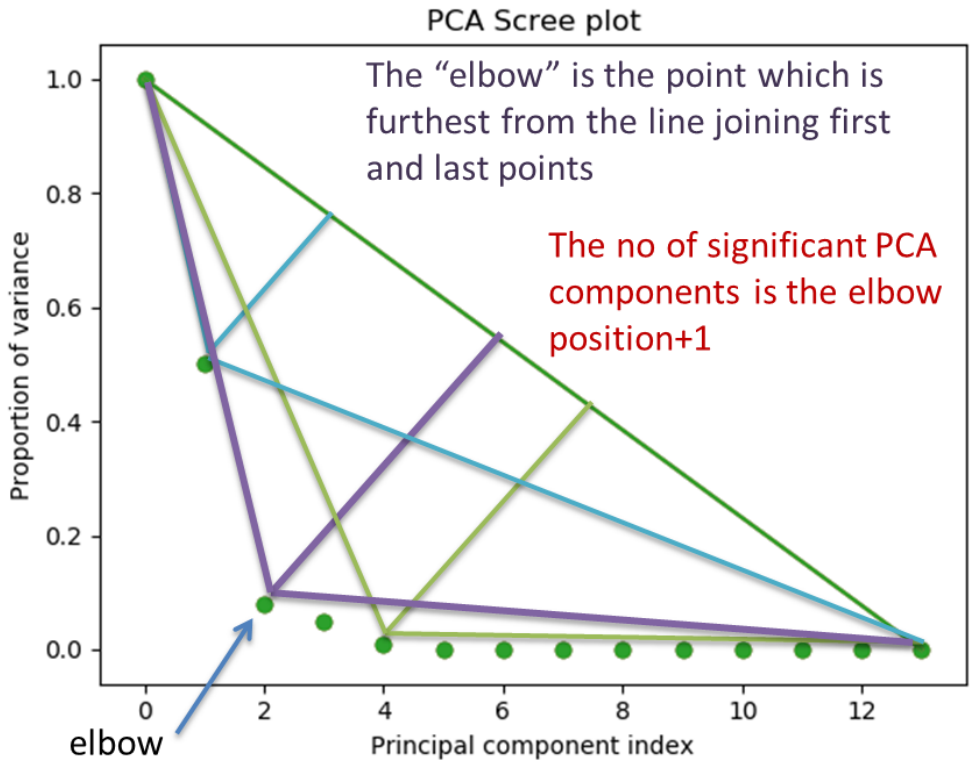
Basic Definitions

Given the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, a scree plot can show:

1. **Raw Singular Values:**
$$(\sigma_i \text{ vs } i)$$
2. **Squared Singular Values:**
$$(\sigma_i^2 \text{ vs } i)$$
3. **Explained Variance Ratio (EVR):**
$$\left(\frac{\sigma_i^2}{\sum_{j=1}^{r} \sigma_j^2} \text{ vs } i \right)$$
4. **Cumulative EVR:**
$$\left(\frac{\sum_{j=1}^{i} \sigma_j^2}{\sum_{j=1}^{r} \sigma_j^2} \text{ vs } i \right)$$

You can choose one or more values to inspect.





Kaiser Criterion

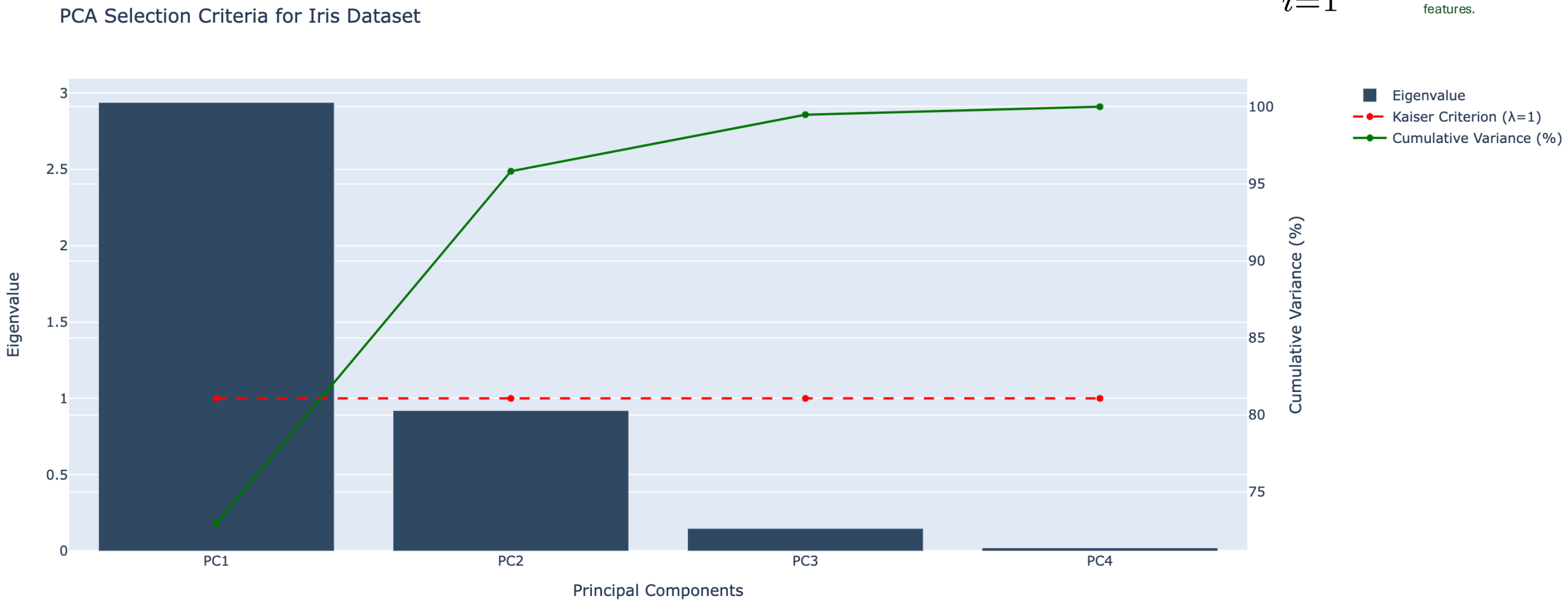
The criterion separates those PCs that explain more or less than their fair share.

Keep all principal components with eigenvalues greater than 1.0.

Data must be standardized.

$$\sum_{i=1}^{p} \lambda_i = p$$

p is the number of features.



Loadings and Scores: Project onto Columns of **V**

Because of the special nature of the vectors in **V**, we would like to project the original data onto them.

You can think of this as a change of basis or coordinate system.

Recall that a projection (or dot product) is given by a matrix product.

A **loading** is just a value within **V**:

$$\text{loading}_{ji} = v_{ji}$$

The projection, using these loadings, gives the **scores**:

$$\underbrace{\mathbf{XV}}_{\text{PC scores}} = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \cdots \quad \mathbf{X}_p] \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1p} \\ v_{21} & v_{22} & \cdots & v_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ v_{p1} & v_{p2} & \cdots & v_{pp} \end{bmatrix}$$

For a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, the scores are:

$$\text{Scores} = \mathbf{XV} = \mathbf{U\Sigma}$$

For a single sample k and component i:

$$\text{score}_{ki} = \sum_{j=1}^{p} x_{kj} v_{ji}$$

Principal Components

For a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$:

$$\mathbf{X} = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \cdots \quad \mathbf{X}_p]$$

where:

- Each $\mathbf{X}_j$ is a column vector in $\mathbb{R}^n$ (one variable measured across n samples)
- $\mathbf{X}_j = [x_{1j} \quad x_{2j} \quad \cdots \quad x_{nj}]^\top$

Then the ith principal component is:

$$\mathrm{PC}_i = \sum_{j=1}^{p} v_{ji} \mathbf{X}_j$$

Principal Components (PCs)

Explicitly writing out what this means:

$$\mathrm{PC}_i = \begin{bmatrix} \mathrm{PC}_{i,1} \\ \mathrm{PC}_{i,2} \\ \vdots \\ \mathrm{PC}_{i,n} \end{bmatrix} = v_{1i} \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{bmatrix} + v_{2i} \begin{bmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{bmatrix} + \cdots + v_{pi} \begin{bmatrix} x_{1p} \\ x_{2p} \\ \vdots \\ x_{np} \end{bmatrix}$$

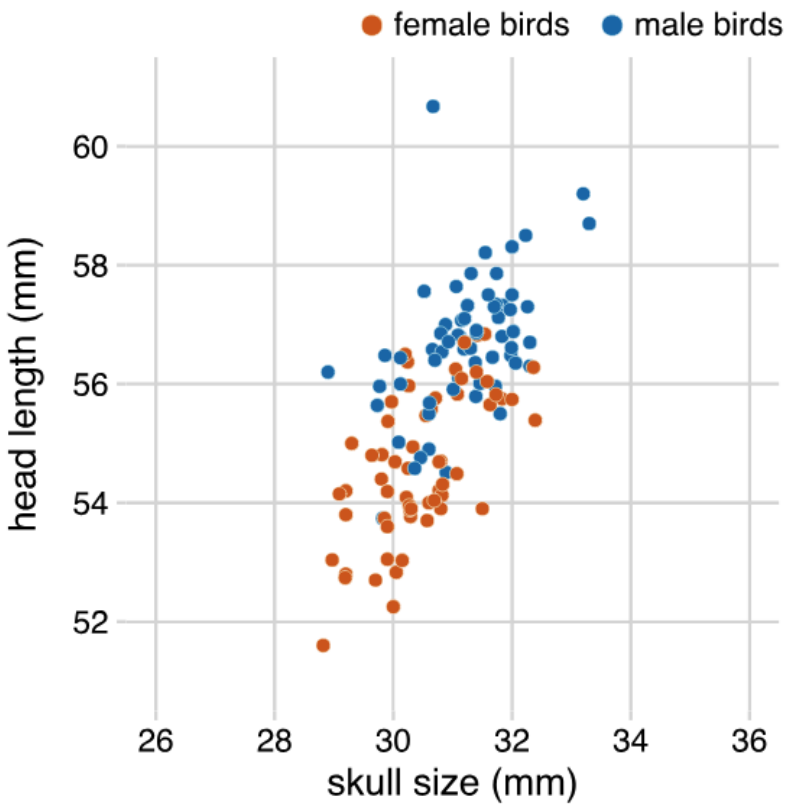
So for any particular sample k:

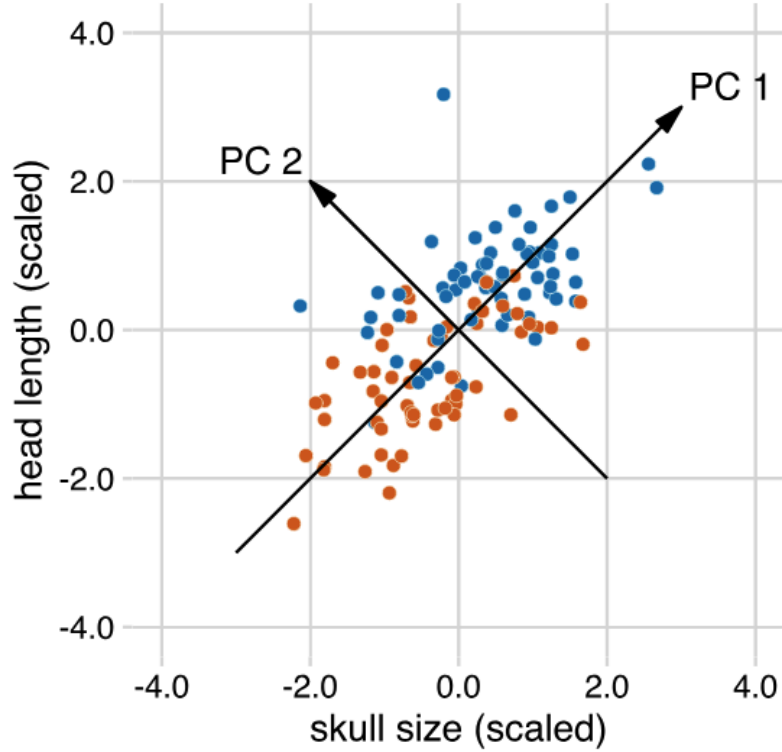
$$\mathrm{PC}_{i,k} = v_{1i}x_{k1} + v_{2i}x_{k2} + \cdots + v_{pi}x_{kp}$$

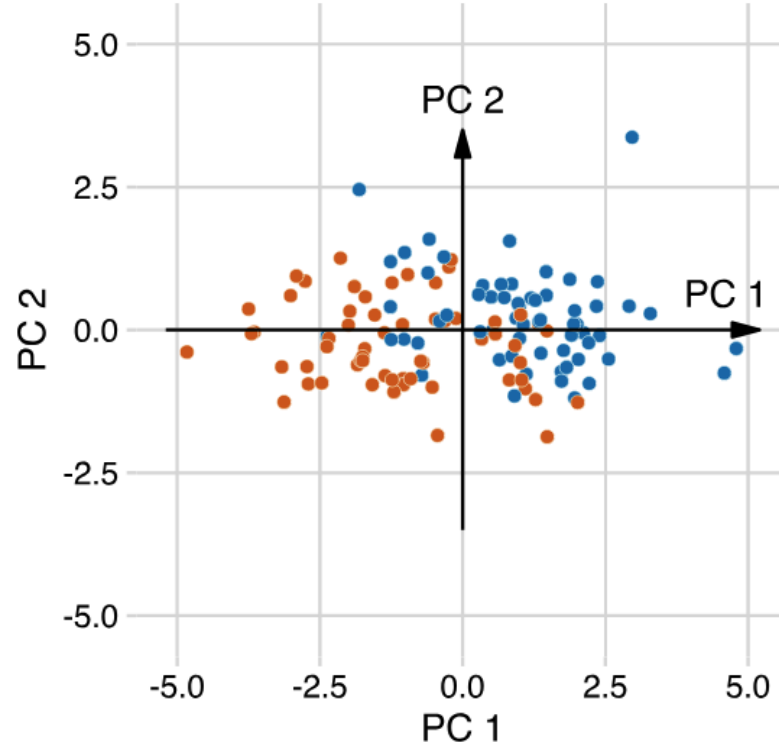
$$x_j^{(n\times 1)} = \sum_{i=1}^{k} v_{ji}^{(1\times 1)} \mathrm{PC}_i^{(n\times 1)} \qquad [\mathrm{PC}_1 \;\; \mathrm{PC}_2 \;\; \cdots \;\; \mathrm{PC}_k] = U_k \Sigma_k$$

The PCs are truncated UΣ:

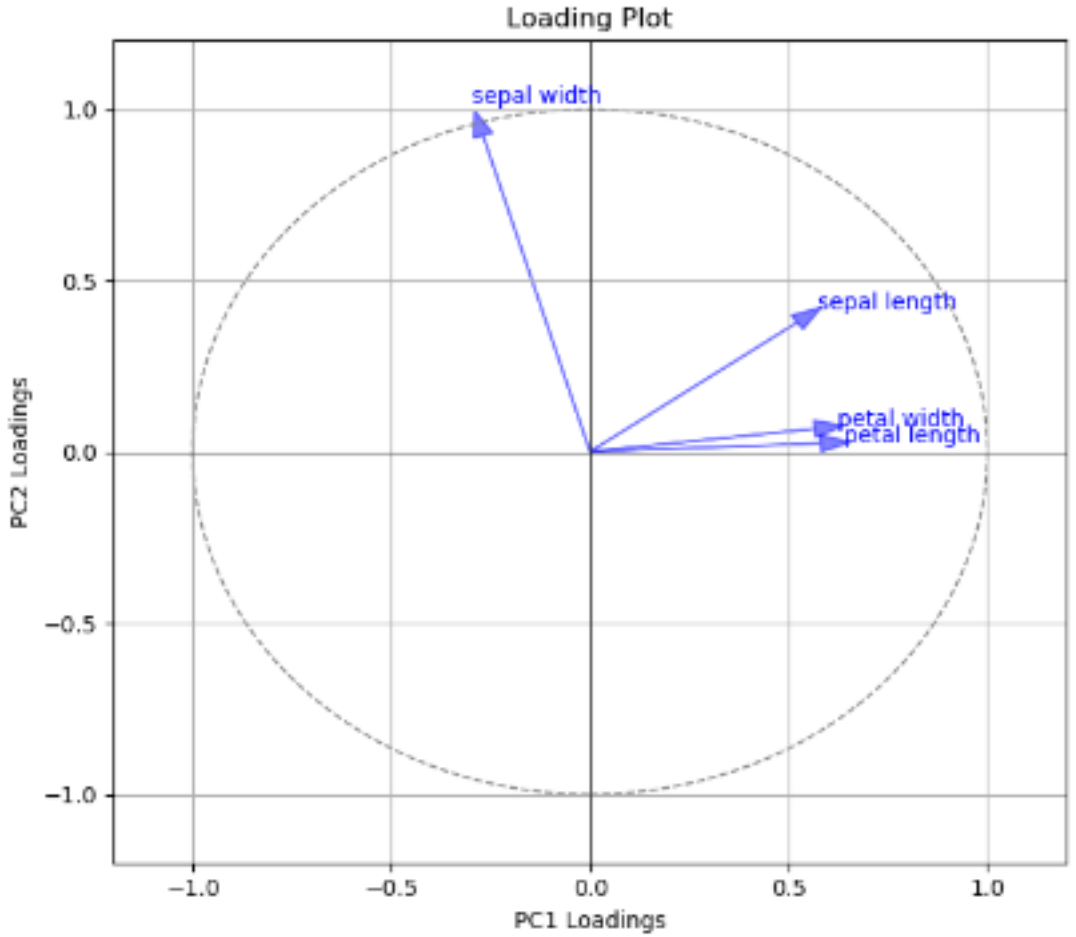
$$x_j^{(1\times n)} \approx \begin{bmatrix} v_{j1} & v_{j2} & \cdots & v_{jk} \end{bmatrix}^{(1\times k)} U_k \Sigma_k^{(k\times n)}$$


female birds
male birds
head length (mm)
skull size (mm)
60
58
56
54
52
26
28
30
32
34
36


PC 1
PC 2
head length (scaled)
skull size (scaled)
4.0
2.0
0.0
-2.0
-4.0


PC 2
PC 1
PC 2
PC 1
5.0
2.5
0.0
-2.5
-5.0

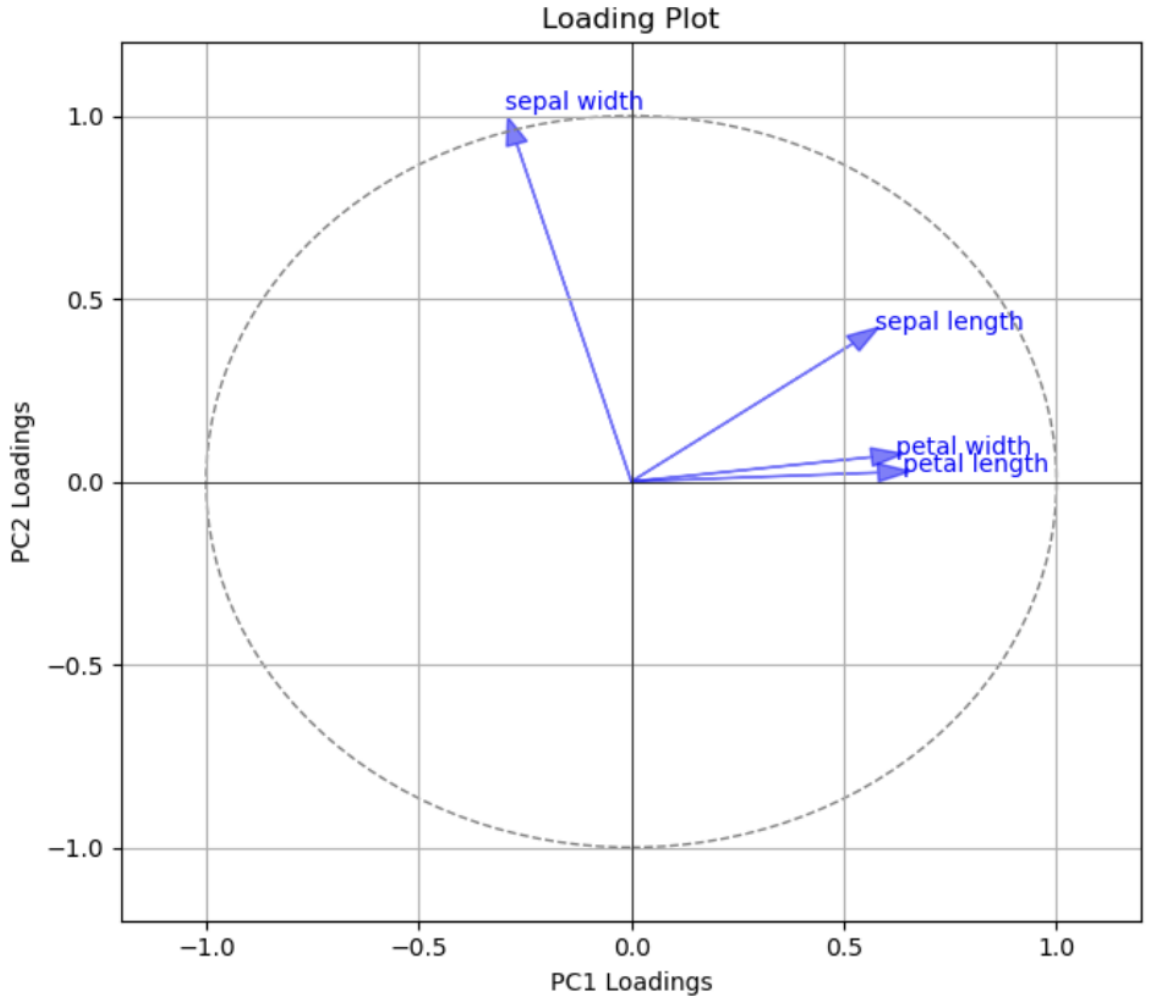
Loading Plot

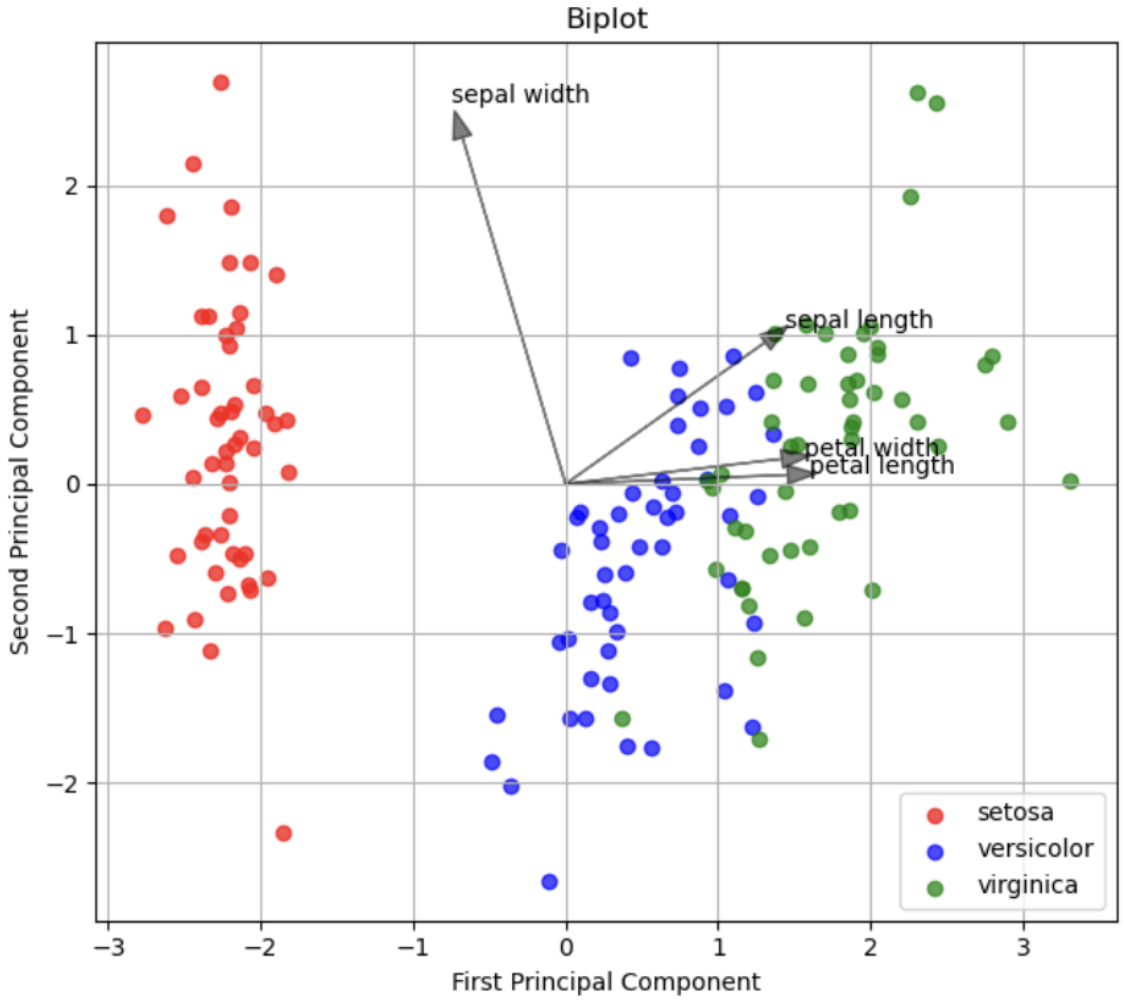


$$x_j = \sum_{i=1}^{p} v_{ji} \mathrm{PC}_i$$

$$\begin{bmatrix} \text{sepal length} \\ \text{sepal width} \\ \text{petal length} \\ \text{petal width} \end{bmatrix}_{4\times 1} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ v_{31} & v_{32} \\ v_{41} & v_{42} \end{bmatrix}_{4\times 2} \begin{bmatrix} \mathrm{PC}_1 \\ \mathrm{PC}_2 \end{bmatrix}_{2\times 1}$$

Score Plots and Loading Plots: Biplots





Loading: weight/coefficient that describes how much each original variable contributes to each principal component.

Score: coordinates of a data point when projected onto the principal component axes.

Biplot: visualization that simultaneously displays both the scores (data points) and loadings (variable vectors) to show relationships between observations, variables, and principal components.

PCA for Dimensionality Reduction

Why Dimensionality Reduction?

Computational Benefits

- Reduces computational complexity
- Decreases storage requirements
- Speeds up model training
- Helps avoid overfitting

Visualization Benefits

- Enables 2D/3D plotting
- Facilitates data exploration
- Helps communicate findings
- Supports pattern recognition

Data Understanding

- Removes redundant features
- Identifies key patterns
- Reduces noise in data
- Reveals hidden structures

Practical Applications

- Image compression
- Gene expression analysis
- Text document clustering
- Recommender systems

PCA for Visualization

DR for Visualization: A Key Use Case

Why Visualize?

- Human perception limited to 2D/3D
- Complex relationships become apparent
- Outliers easier to detect
- Clusters become visible

Visualization Process

- Select appropriate DR technique
- Choose target dimensionality (usually 2D/3D)
- Apply transformation
- Create visual representation

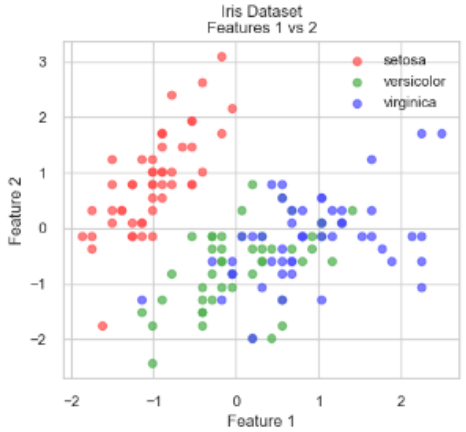
Common Techniques

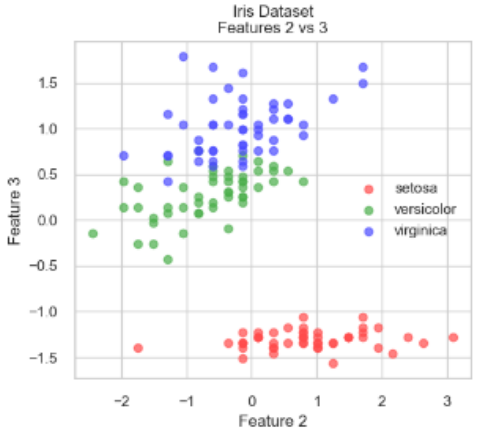
- PCA (linear, preserves variance)
- t-SNE (non-linear, preserves local structure)
- UMAP (non-linear, preserves global structure)
- MDS (preserves distances)

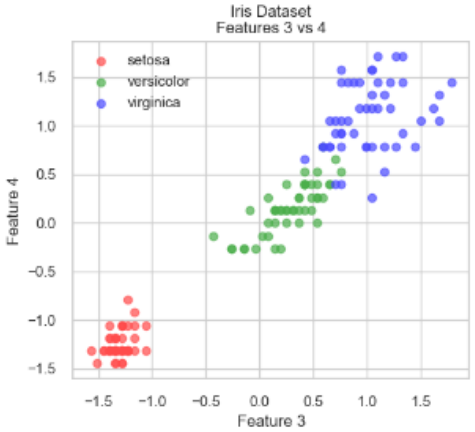
Interpretation Tips

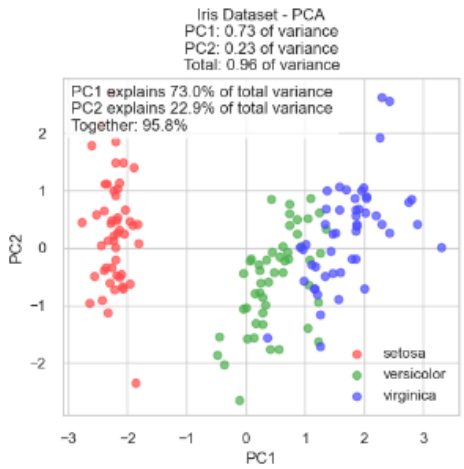
- Consider what's preserved vs. lost
- Use multiple techniques
- Add context (colors, labels, etc.)
- Validate patterns with statistics

Iris
4D

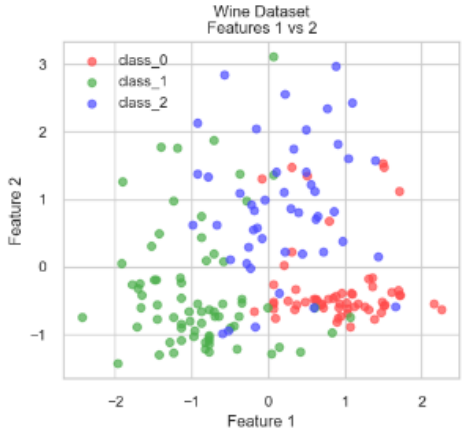


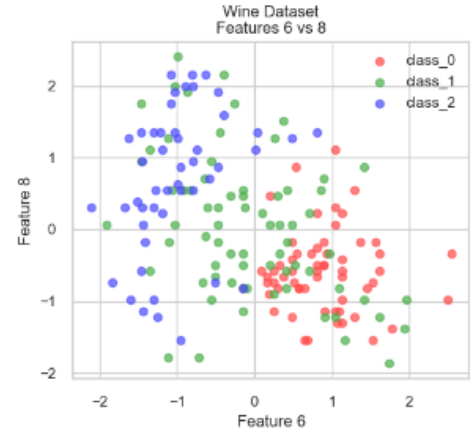


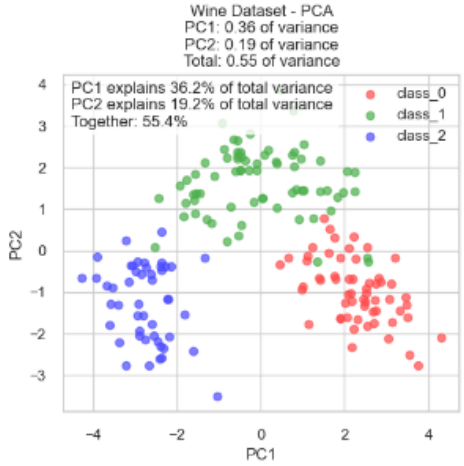




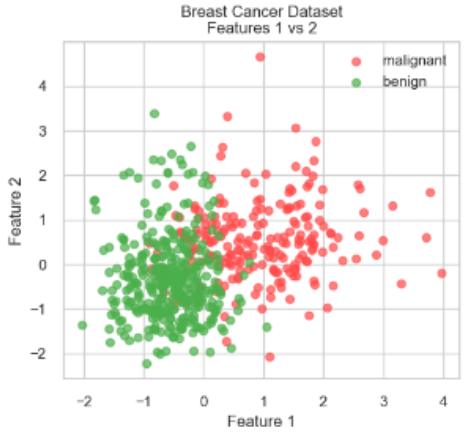
Wine
13D

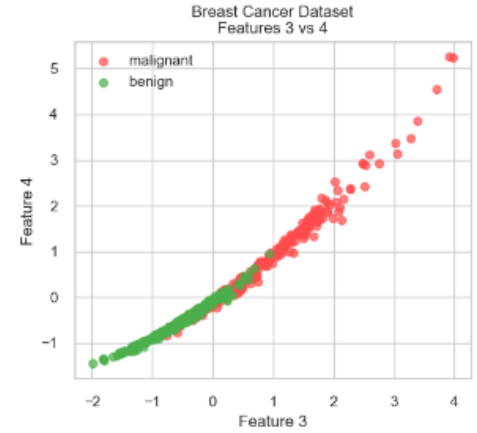


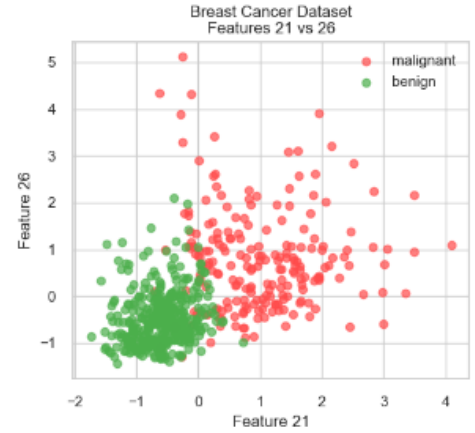


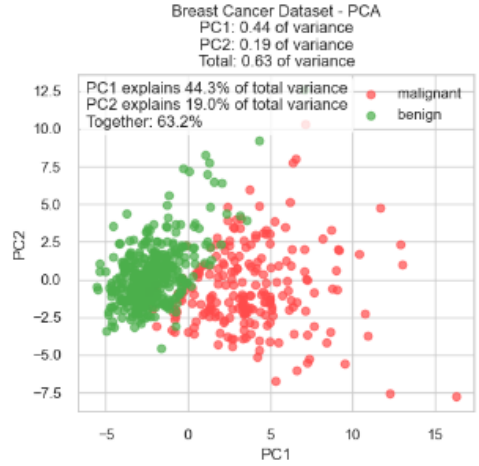


Breast
Cancer
30D

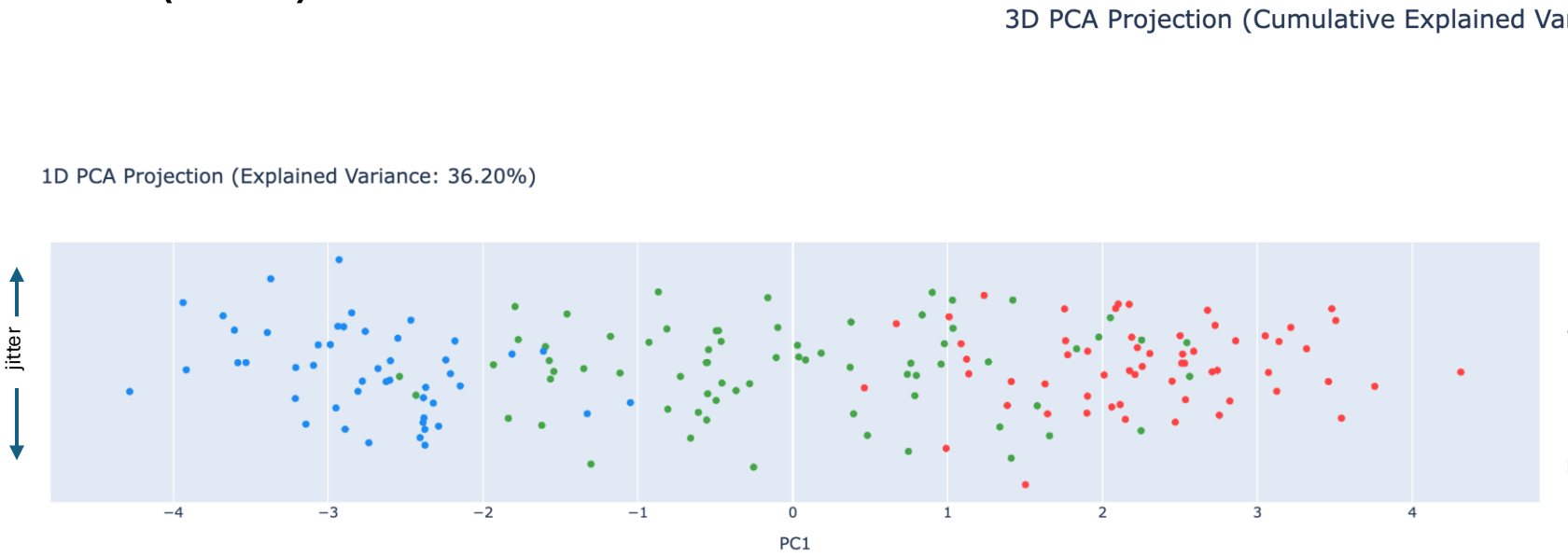


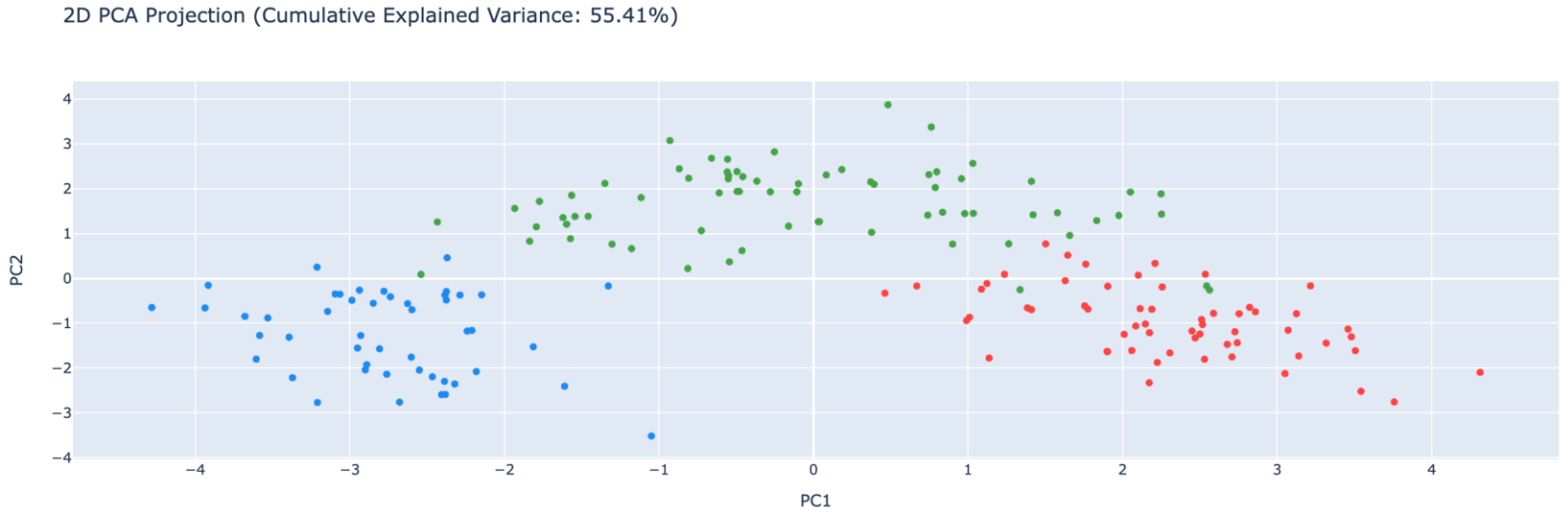


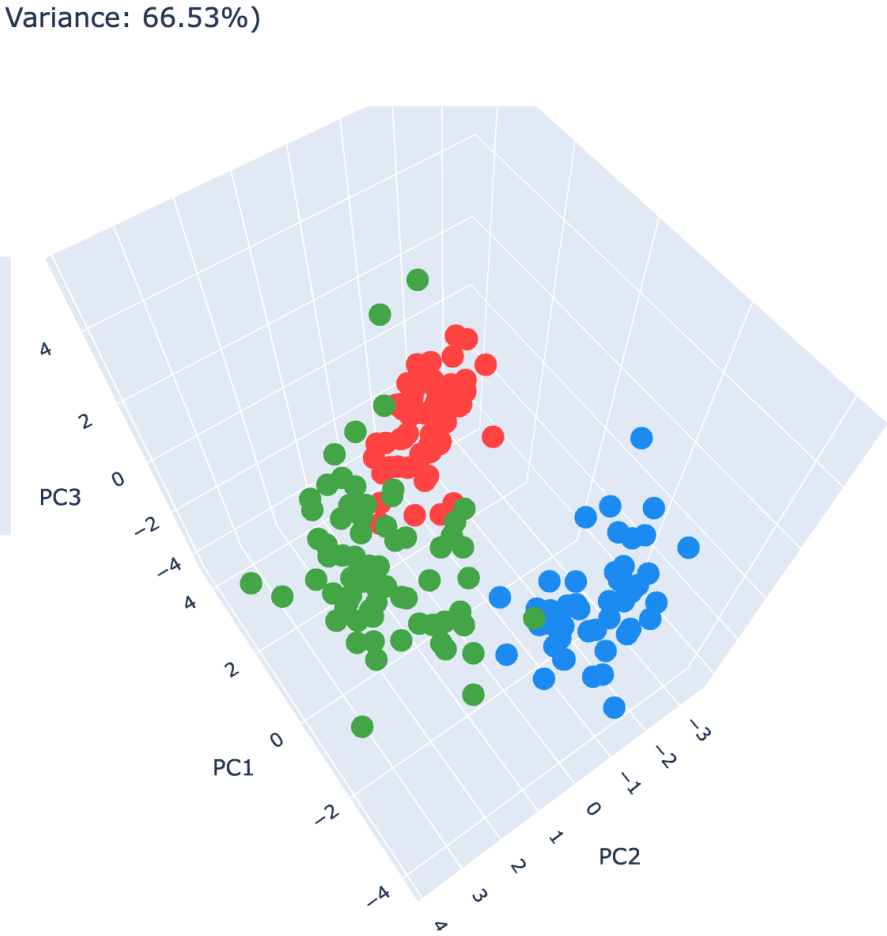




Wine (13D)







The explained variance indicates how much information the visualization might be losing.