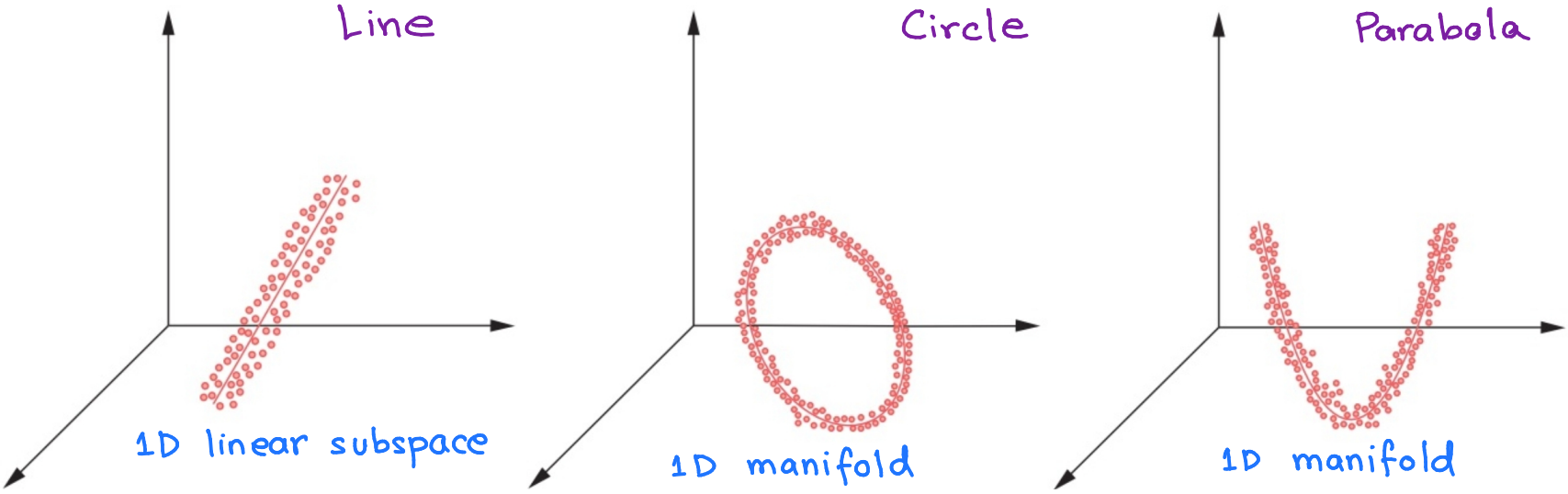
# Dimensionality Reduction

- In unsupervised learning, we have seen clustering.
- In this lecture, we will look at dimensionality reduction
- In many practical applications, the input data $\underline{x} \in \mathbb{R}^p$ is a very high-dimensional, however, the intrinsic dimensionality may be quite small




In all three cases, the intrinsic dimensionality of data is 1

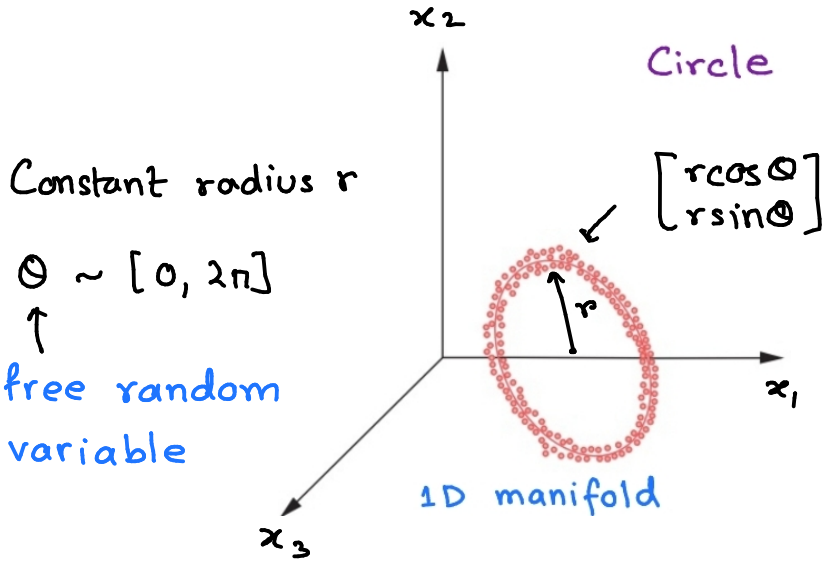
# Intrinsic Dimensionality

- A data set $\{\underline{x}^{(i)}\}_{i=1}^{N}$, with $\underline{x} \in \mathbb{R}^P$, is said to have intrinsic dimensionality $M \leq P$, if the dataset can be described effectively in terms of 'M' free random variables

$$\underline{x} = g(\underline{u})$$

$\underline{x} \in \mathbb{R}^P$, $\underline{u} \in \mathbb{R}^M$

Example




The data lies along the circumference of a circle of radius r and a single free parameter $\theta$ suffices to describe the data

Intrinsic dimension = 1

# Intrinsic Dimensionality

- An important concern in ML is learning from high-dimensional data $x$
- Success of ML, in particular deep learning, is due to its capability of

> learning a useful representation of high-dimensional data

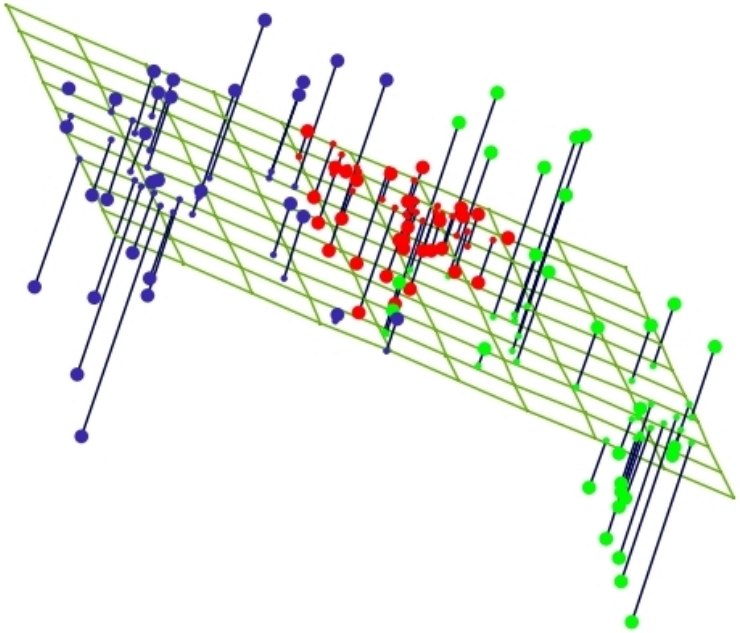
- One of the goals of unsupervised learning:

  Learning a lower-dimensional subspace for encoding high-dimensional data set

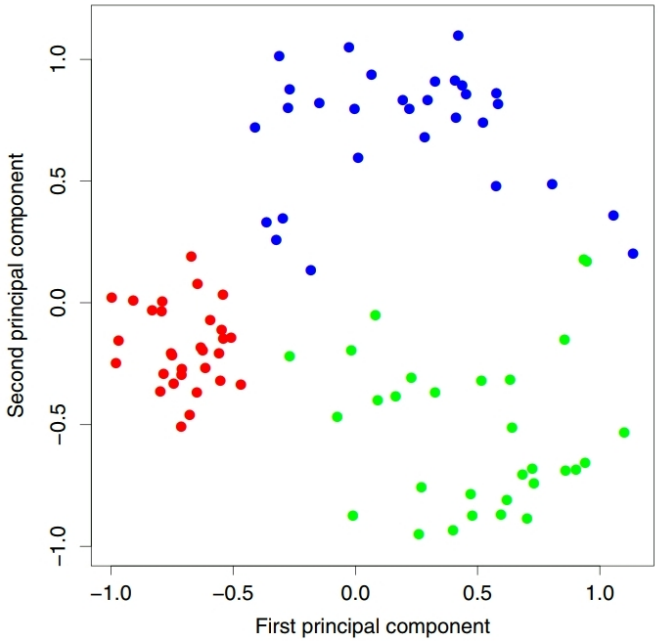
- Idea of dimensionality reduction: Map data to a lower dimensional space
  - Save computational time in modelling high-dimensional data
  - Visualization in 2-dimensions can offer insights
  - Reduce overfitting and achieve better generalization

# Linear Dimensionality Reduction

- We will introduce linear dimensionality reduction using Principal Component Analysis (PCA)
- PCA is also known as Karhunen-Loève (KL) transform
  - It falls under linear dimensionality reduction techniques



Projection on a linear subspace



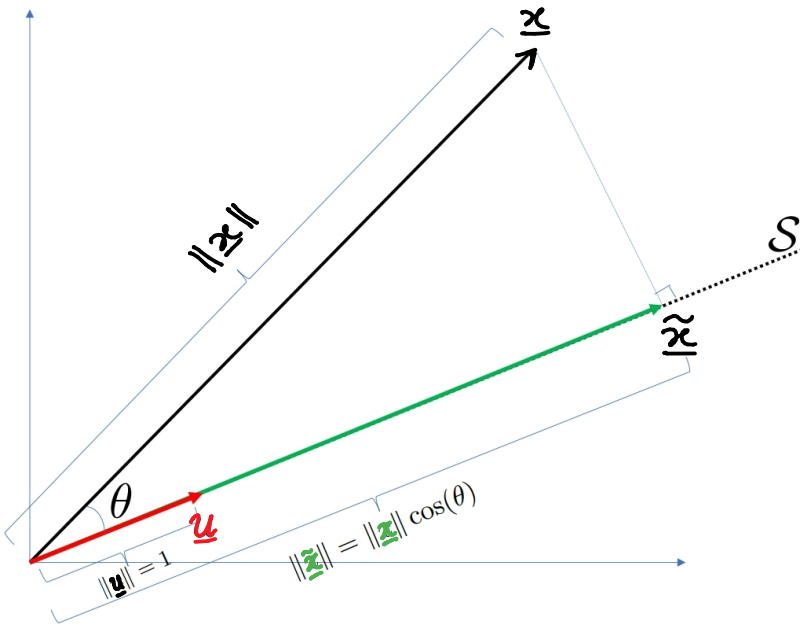

3D space

PCA →

2D-space

# Idea of projection

- Consider projection onto 1-D subspace (a line)




- Subspace $S$ is the line along the unit vector $\underline{u}$
  - $\underline{u}$ is the basis of $S$: Any point in $S$ can be written as $z\underline{u}$ for some scalar $z$

- Projection of vector $\underline{x}$ on $S$ is denoted by $\tilde{\underline{x}} = \text{Proj}_S(\underline{x})$
- Recall that: $\underline{x}^T \underline{u} = \|\underline{x}\| \|\underline{u}\| \cos(\theta) = \|\underline{x}\| \cos\theta = \|\tilde{\underline{x}}\|$ ($\|\underline{u}\| \to 1$)
- $\tilde{\underline{x}} = \text{Proj}_S(\underline{x}) = \underbrace{\underline{x}^T \underline{u}}_{\text{length of projection}} \cdot \underbrace{\underline{u}}_{\text{direction of projection}} = \|\tilde{\underline{x}}\| \, \underline{u}$

# Idea of projection

- How to project onto a K-dimensional subspace?
  - Idea: Choose an orthonormal bases $\{\underline{u}_1, \underline{u}_2, \ldots, \underline{u}_K\}$ for S
  - Project onto each unit vector individually (as in previous slide) and sum together the projections

- Mathematically, the projection is given as:

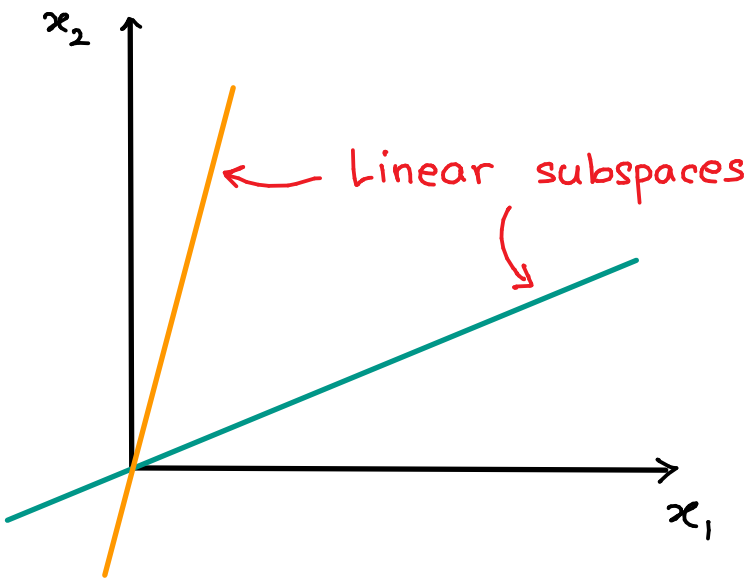
$$\tilde{\underline{x}} = \text{Proj}_S(\underline{x}) = \sum_{i=1}^{K} z_i \underline{u}_i \quad \text{where} \quad z_i = \underline{x}^T \underline{u}_i$$

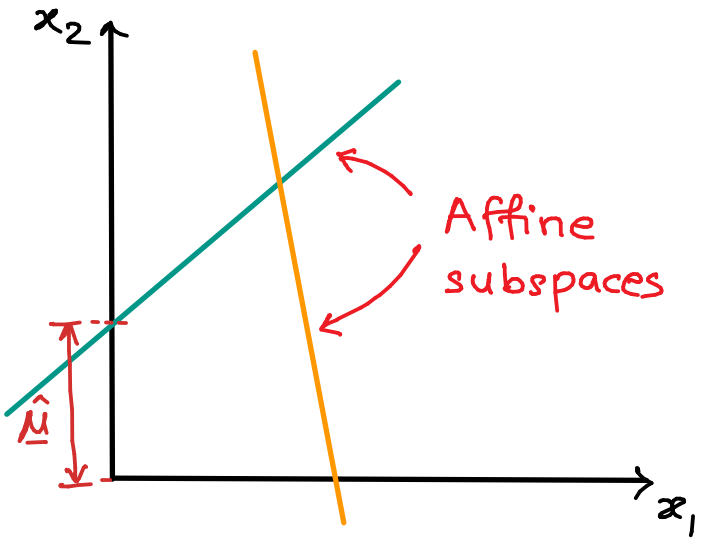
- In vector form:

$$\tilde{\underline{x}} = \text{Proj}_S(\underline{x}) = \underline{\underline{U}}\,\underline{z} = \begin{bmatrix} | & | & & | \\ \underline{u}_1 & \underline{u}_2 & \cdots & \underline{u}_K \\ | & | & & | \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_K \end{bmatrix}, \quad \text{where} \quad \underline{z} = \underline{\underline{U}}^T \underline{x}$$

# Projection onto an affine subspace

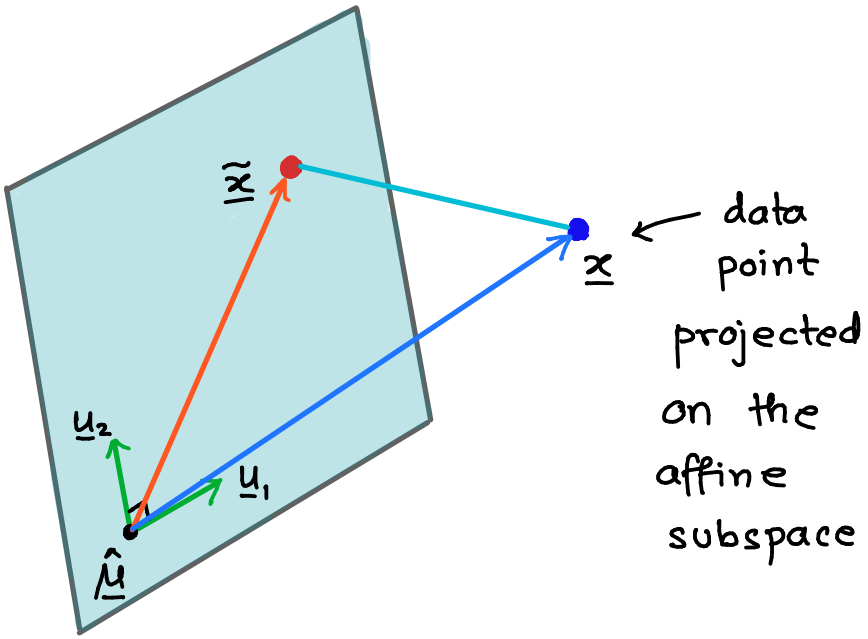
- So far, we have assumed a subspace that passes through zero
- However, the subspaces that we want to project onto can also be affine subspaces, which need not pass through zero




The affine subspaces can have an arbitrary origin $\hat{\underline{\mu}}$

# Projection onto an affine subspace




$$\tilde{x} = \text{Proj}_S(x)$$
$$= U z + \hat{\mu}$$
$$= z_1 u_1 + z_2 u_2 + \hat{\mu}$$

$$z = U^T (x - \hat{\mu})$$

The affine subspace has an origin $\hat{\mu}$

- $\tilde{x}$ is called the reconstruction of $x$
- $z$ is its feature / code
- If all the data points $x$ lie close to the subspace, we could approximate $x$ with its reconstructions $\tilde{x}$

$$x \approx U z + \hat{\mu}$$

## How to choose a good subspace?

- We want to choose a subspace $S$ which is low-dimensional compared to the dimension of the input space
- How to choose such a subspace $S$?
  - We need to find appropriate $\hat{\underline{\mu}}$ and the orthogonal bases $\underline{\underline{U}}$
  - origin $\hat{\underline{\mu}}$ can be set equal to the mean of the dataset
- To find $\underline{\underline{U}}$, one of the two equivalent criteria could be followed:
  - Minimize the reconstruction error:

$$\underset{\underline{\underline{U}}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \| \underline{x}^{(i)} - \tilde{\underline{x}}^{(i)} \|_2^2$$

  - Maximize the variance of reconstructions: Find a subspace where the data has the most variability

$$\underset{\underline{\underline{U}}}{\arg\max} \frac{1}{N} \sum_{i=1}^{N} \| \underline{x}^{(i)} - \tilde{\underline{x}}^{(i)} \|_2^2$$

# Principal Component Analysis

- Choosing a subspace to maximize the projected variance, or minimize the reconstruction error, is called PCA

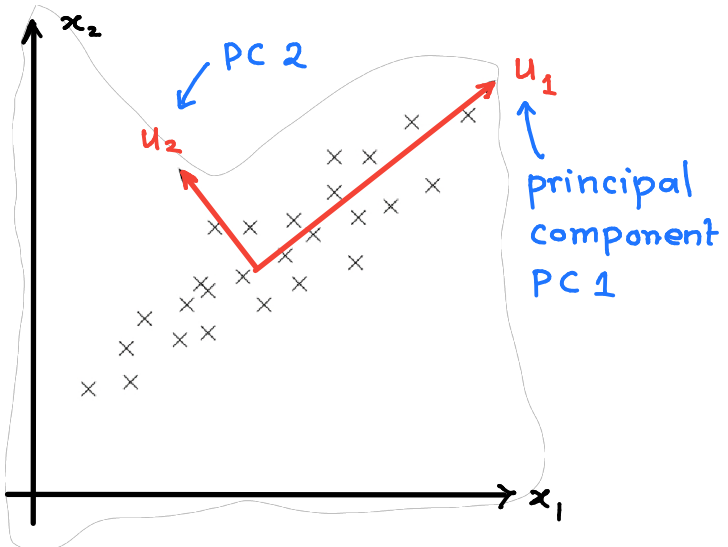
- Consider the sample covariance matrix:

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (x^{(i)} - \hat{\mu})(x^{(i)} - \hat{\mu})^T$$

- $\hat{\Sigma}$ is symmetric and Positive semi-definite (PSD)

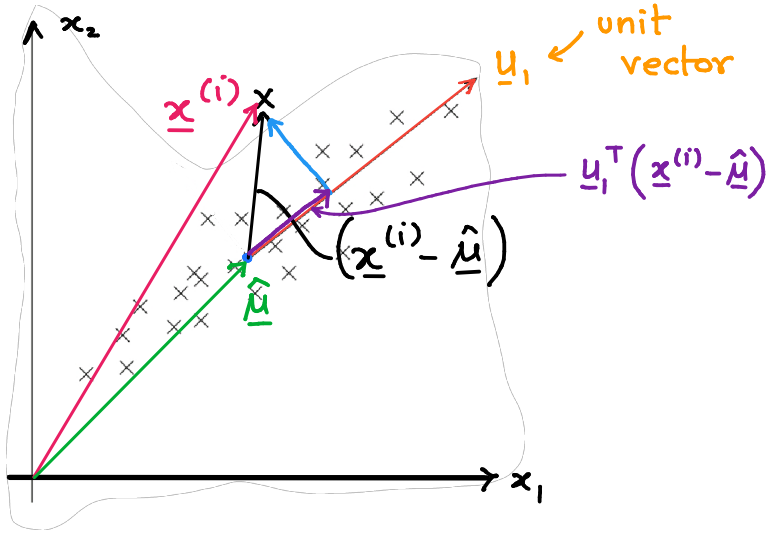
- The optimal PCA subspace is spanned by the top 'M' eigenvectors of $\hat{\Sigma}$

- These eigenvectors are called principal components or principal directions, much like the principal axes of an ellipse



# Derivation of PCA

- Let us consider the simplest case of finding a 1-D subspace
  - The goal then is to find a single direction represented by unit vector $\underline{u}_1$




- Lets maximize the projected variance

$$J(\underline{u}_1) = \frac{1}{N}\sum_{i=1}^{N}\left(\underline{u}_1^T(\underline{x}^{(i)} - \hat{\underline{\mu}})\right)^2$$

$$= \frac{1}{N}\sum_{i=1}^{N} \underline{u}_1^T (\underline{x}^{(i)}-\hat{\underline{\mu}})(\underline{x}^{(i)}-\hat{\underline{\mu}})^T \underline{u}_1$$

$$= \underline{u}_1^T \hat{\underline{\underline{\Sigma}}}\, \underline{u}_1$$

- So the optimization task is:

$$\underline{u}_1 = \arg\max_{\underline{u}} \; \underline{u}^T \hat{\underline{\underline{\Sigma}}}\, \underline{u}$$

$$\text{s.t.} \quad \underline{u}^T\underline{u} = 1$$

Lagrangian: $L(\underline{u}, \lambda) = \underline{u}^T \hat{\underline{\underline{\Sigma}}}\, \underline{u} - \lambda(\underline{u}^T\underline{u} - 1)$

Take gradient and set to zero:

$$\hat{\underline{\underline{\Sigma}}}\, \underline{u} = \lambda \underline{u}$$

($\lambda$: eigenvalue; $\underline{u}$: eigenvector)

$\therefore$ Principal direction $\underline{u}_1$ is an eigenvector

- Since $\hat{\underline{\underline{\Sigma}}}$ is symmetric and PSD, all eigenvalues are real and non-negative: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$

- The 2nd principal component $\underline{u}_2$ is selected such that:
  - (a) $\underline{u}_2$ is orthogonal to $\underline{u}_1$
  - (b) $\underline{u}_2$ maximizes the variance after projecting the data onto the direction of $\underline{u}_2$
  - (c) The 2nd principal component (or direction) is the eigenvector corresponding to the 2nd largest eigenvalue of $\hat{\underline{\underline{\Sigma}}}$, $\lambda_2$

- Similar arguments can be used to show that the 'm'th principal component is the 'm'th eigenvector of $\hat{\underline{\underline{\Sigma}}}$

- The process continues until M principal components (corresponding to the M largest eigenvalues)

# PCA decorrelates features

- The features (or code) are decorrelated by PCA

$$\begin{aligned}
\mathrm{Cov}(\underline{z}) &= \mathrm{Cov}\left(\underline{\underline{U}}^T(\underline{x}-\hat{\underline{\mu}})\right) \\
&= \underline{\underline{U}}^T \mathrm{Cov}(\underline{x})\, \underline{\underline{U}} \\
&= \underline{\underline{U}}^T \hat{\underline{\underline{\Sigma}}}\, \underline{\underline{U}} \\
&= \underline{\underline{U}}^T \underline{\underline{Q}}\, \underline{\underline{\Lambda}}\, \underline{\underline{Q}}^T \underline{\underline{U}} \\
&= [\underline{\underline{I}}\ \ \underline{\underline{0}}]\, \underline{\underline{\Lambda}} \begin{bmatrix} \underline{\underline{I}} \\ \underline{\underline{0}} \end{bmatrix} \\
&= \text{top left } M \times M \text{ block of } \underline{\underline{\Lambda}}
\end{aligned}$$

Spectral decomposition

$$\underset{P\times P}{\hat{\underline{\underline{\Sigma}}}} = \underline{\underline{Q}}\, \underline{\underline{\Lambda}}\, \underline{\underline{Q}}^T$$

$\underline{\underline{Q}}$: eigenvector matrix (orthonormal)

$\underline{\underline{\Lambda}}$: eigenvalues matrix

$$\underset{P\times P}{\underline{\underline{Q}}} = \begin{bmatrix} \underline{\underline{U}} & \underline{\underline{U}}_\perp \\ P\times M & P\times(P-M) \end{bmatrix}$$

- Covariance of feature $\underline{z}$ is diagonal → uncorrelated

# Summary of PCA

- Dimensionality reduction aims to find a low-dimensional representation of the data
- PCA projects the data onto an affine subspace that maximizes projected variance or minimizes the reconstruction error
- The optimal subspace is given by the top M eigenvectors of the sample covariance matrix, corresponding to the M largest eigenvalues
- PCA gives a set of decorrelated features

# Example of data compression