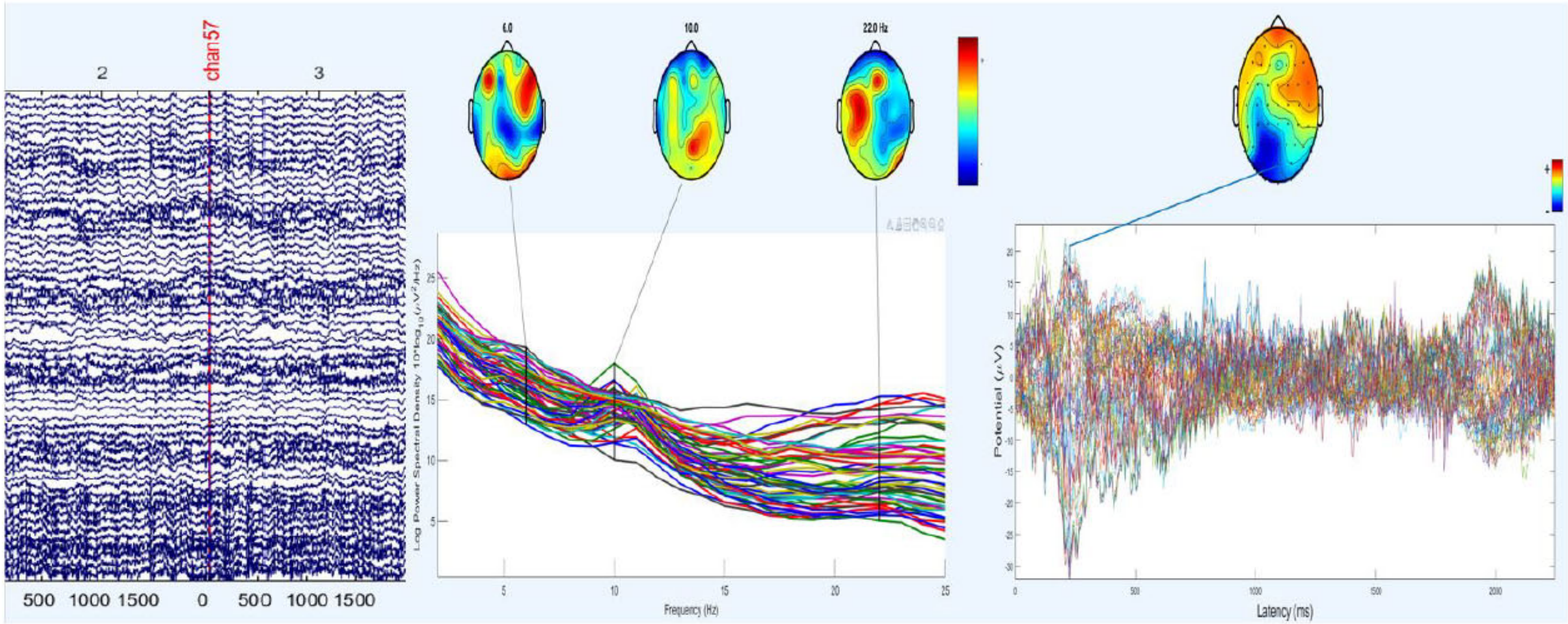
PCA

The slides have been created based on Dr. Sharifi Zarchi's Machine Learning course slides.

High Dimensional Data

- High dimensions has many features.
- EEG signals from the brain, recorded with 56 channels and 3000 time points per trial.



High Dimensional Data

- Social media

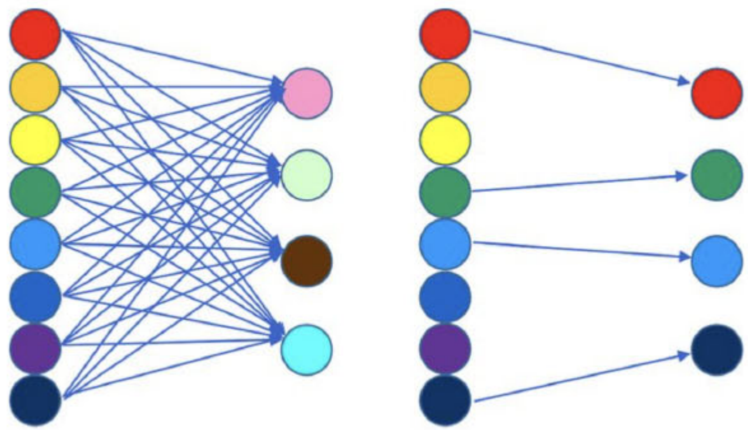
Figure 1: Figure reference

Dimensionality Reduction Benefits

- **Visualization**
  - Project high dimensional data into 2D or 3D.
- **Helps avoid overfitting**
  - Reducing noise by reducing features.
  - Improves accuracy by reducing noise.
- **More efficient use of resources**
  - Time, Memory, CPU

Dimensionality Reduction Techniques

- **Feature Selection**
  - Select a subset from a given feature set.
- **Feature Extraction**
  - A linear or non-linear transform from the original feature space to a lower dimension space.

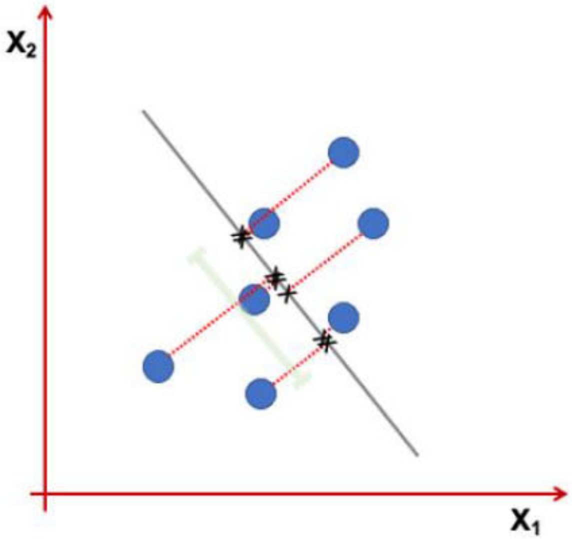


feature extraction feature selection

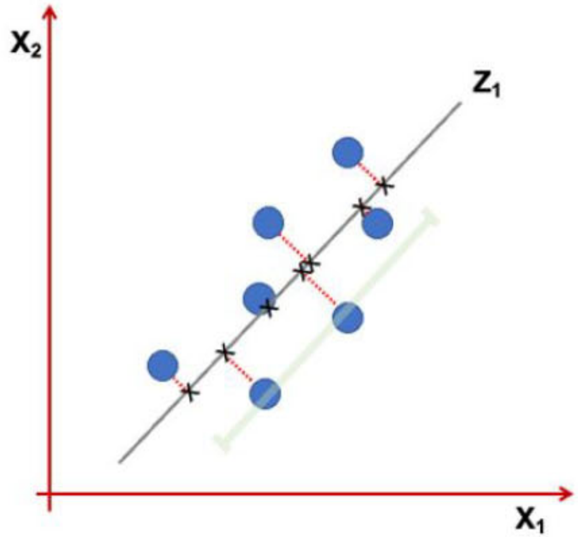
- Maximize retention of **important information** while reducing dimensionality.
- What is **important information**?

Variance of Data

- Maximize retention of **important information** while reducing dimensionality.
- **Information:** Variance of projected data



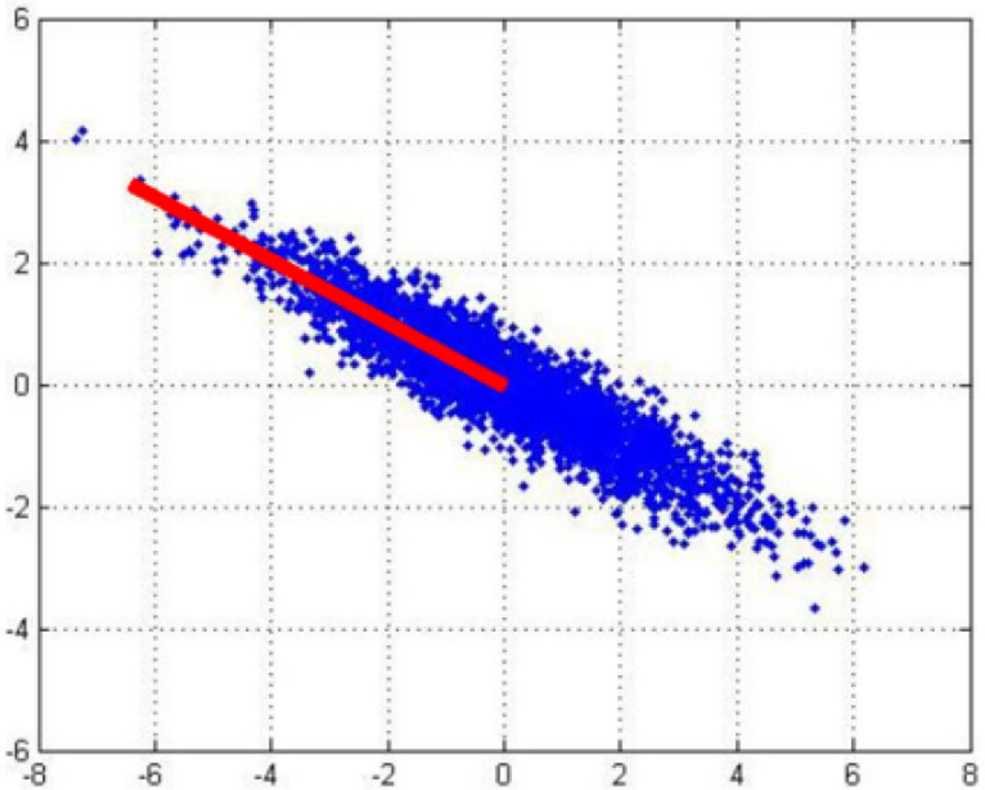
Low Variance



High Variance

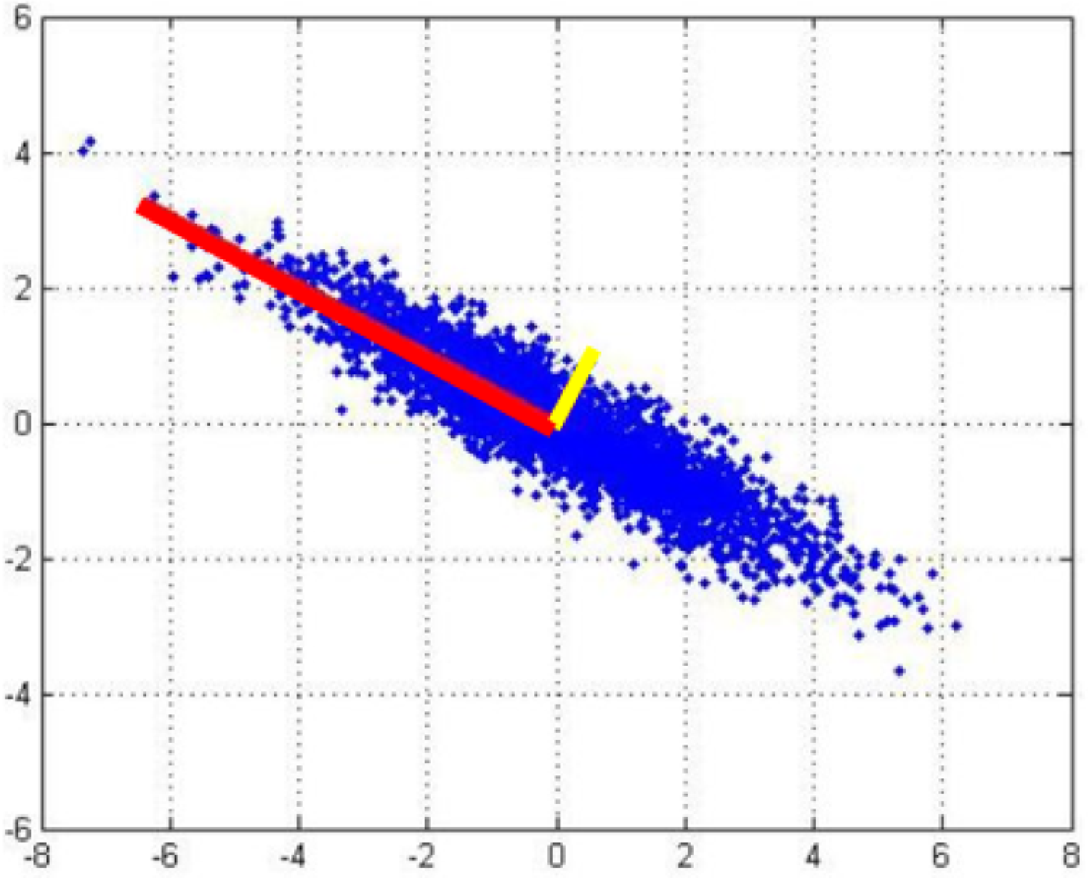
Principal Component Idea

First PCA axis:



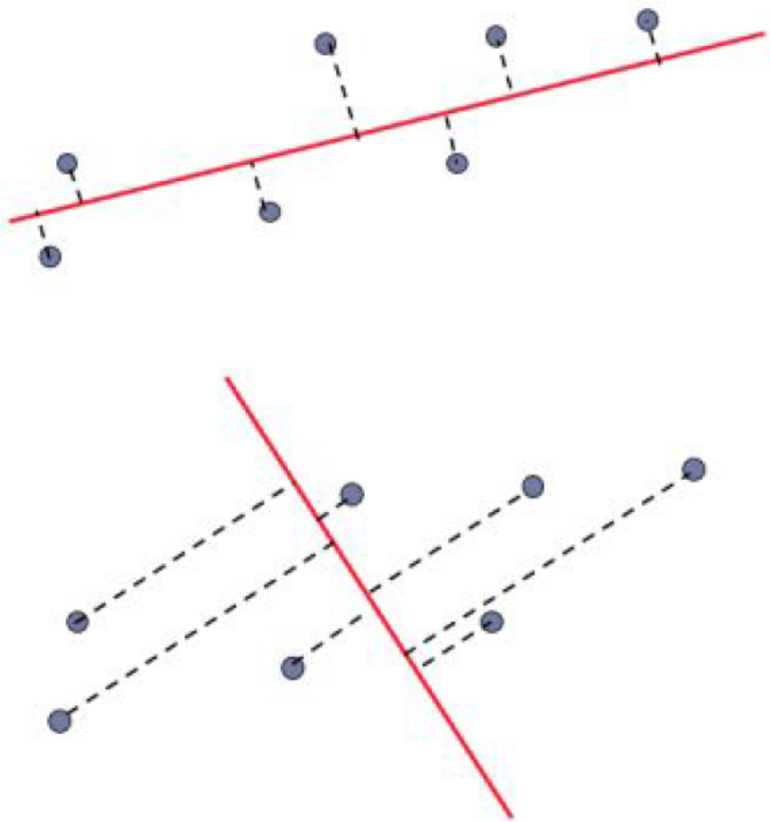
Principal Component Idea

- First and second PCA axes:



Random vs Principal Projection

- Random direction versus principal component:

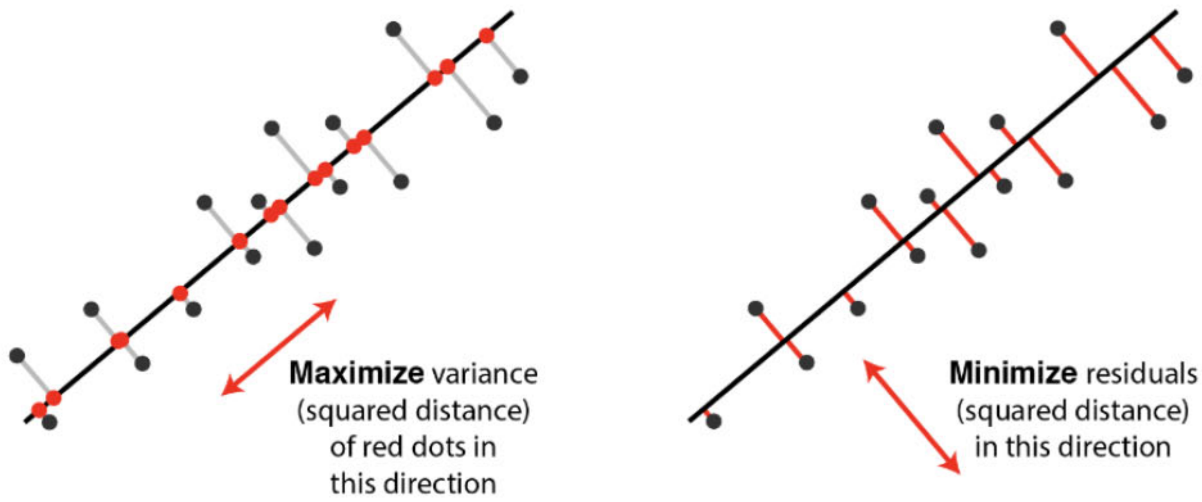


Definition

- **Goal**: reducing the dimensionality of the data while preserving important aspects of the data.
- Suppose $\mathbf{X} = \begin{pmatrix} \mathbf{X}_1^T \\ \vdots \\ \mathbf{X}_N^T \end{pmatrix}_{N \times d} = \begin{array}{cccc} F_1 & F_2 & & F_d \\ \end{array} \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & & & \\ x_{N1} & x_{N2} & \dots & x_{Nd} \end{pmatrix}$
- $\mathbf{X}_{N \times d} \xrightarrow{\text{PCA}} \tilde{\mathbf{X}}_{N \times k}$ with $k \leq d$
- **Assumption**: Data is mean-centered, which is: $\mu_x = \frac{1}{N} \sum_{i=1}^{N} X_i = 0_{d \times 1}$

Orthogonal projection of the data onto a **lower-dimensional** linear **subspace** that:

- **Interpretation 1.** Maximizes variance of projected data.
- **Interpretation 2.** Minimizes the sum of squared distances to the subspace.



Pre processing

- **Mean-center the data.**
  - **Zero**ing out the **mean** of each feature.
- **Scaling to normalize each feature to have variance 1 (an arbitrary step).**
  - Might affect results.
  - It helps when unit of measurements of features are different and some features may be ignored without normalization.

Background

- Before jumping to PCA algorithm, we should be familiar with followings
  - What are eigenvalues and eigenvectors?
  - Sample covariance matrix

What are Eigenvalue and Eigenvector?

- **Eigenvector:** A non-zero vector that multiplies only by a scalar factor when a linear transformation is applied.
- **Eigenvalue:** The scalar factor by which the eigenvector is scaled.
- **Equation** for a $n \times n$ matrix:

$$Av = \lambda v$$

- Where
    - A: A square matrix
    - v: Eigenvector
    - λ: Eigenvalue

How to find eigenvalue and eigenvector ?

- We know that

$$Av = \lambda v$$

- So

$$Av - \lambda v = 0$$

$$(A - \lambda I)v = 0$$

- v can not be zero, so:

$$det(A - \lambda I) = 0$$

- Solve for λ
- Substitute λ back into the equation $Av = \lambda v$ to find v.

Example

- Assume $A = \begin{pmatrix} 4 & -5 \\ 2 & -3 \end{pmatrix}$
- $A - \lambda I = ?$

What is covariance ?

- Covariance is a measure of how much two random features vary together.
- $\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[(Y - \mathbb{E}[Y])(X - \mathbb{E}[X])] = \mathrm{Cov}(Y, X)$
- So covariance is symmetric.
- Such as heights and weights of individuals.

What is covariance matrix ?

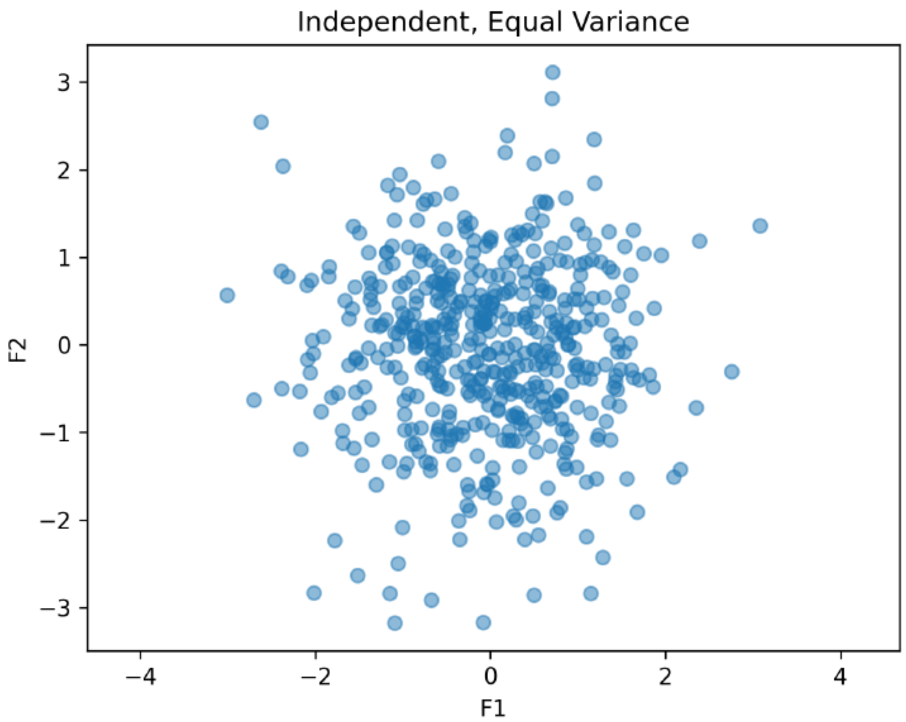
- A **covariance matrix** generalizes the concept of covariance to multiple features.
- For a random vector $\mathbf{F} = [F_1, F_2, \ldots, F_d]$:

$$\Sigma = \begin{pmatrix} \mathrm{Var}(F_1) & \mathrm{Cov}(F_1, F_2) & \cdots & \mathrm{Cov}(F_1, F_d) \\ \mathrm{Cov}(F_2, F_1) & \mathrm{Var}(F_2) & \cdots & \mathrm{Cov}(F_2, F_d) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(F_d, F_1) & \mathrm{Cov}(F_d, F_2) & \cdots & \mathrm{Var}(F_d) \end{pmatrix}$$

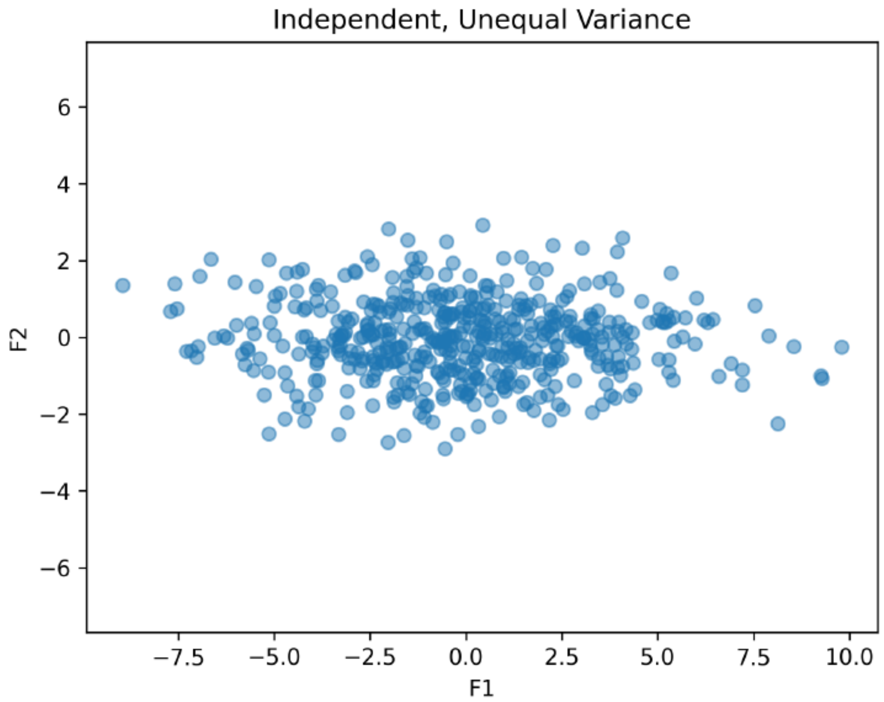
- The diagonal elements are the variances, and off-diagonal elements are covariances.

Covariance Matrix Example

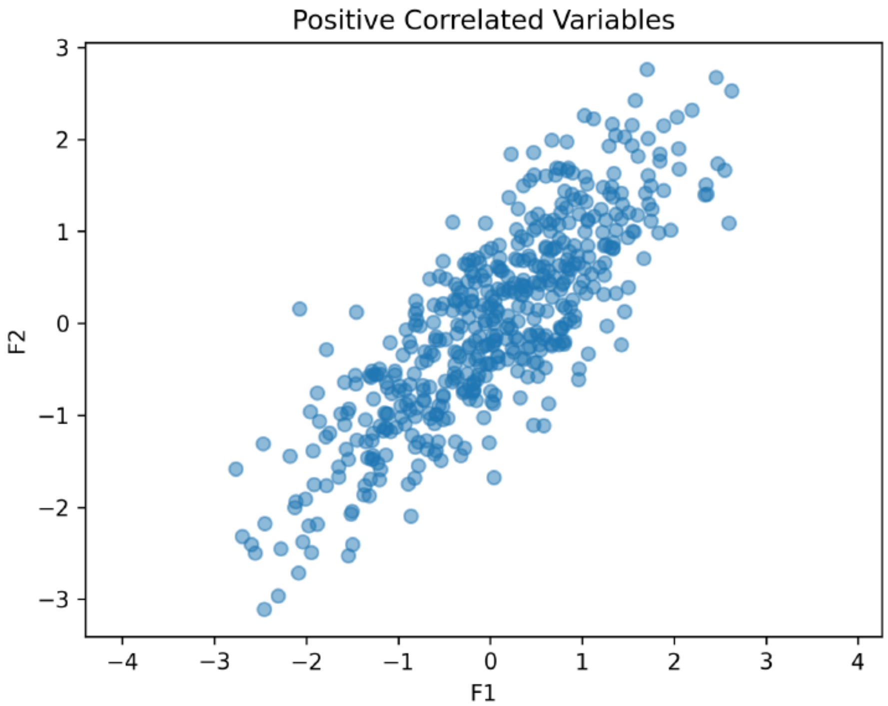
- If $\Sigma = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix}$, then:



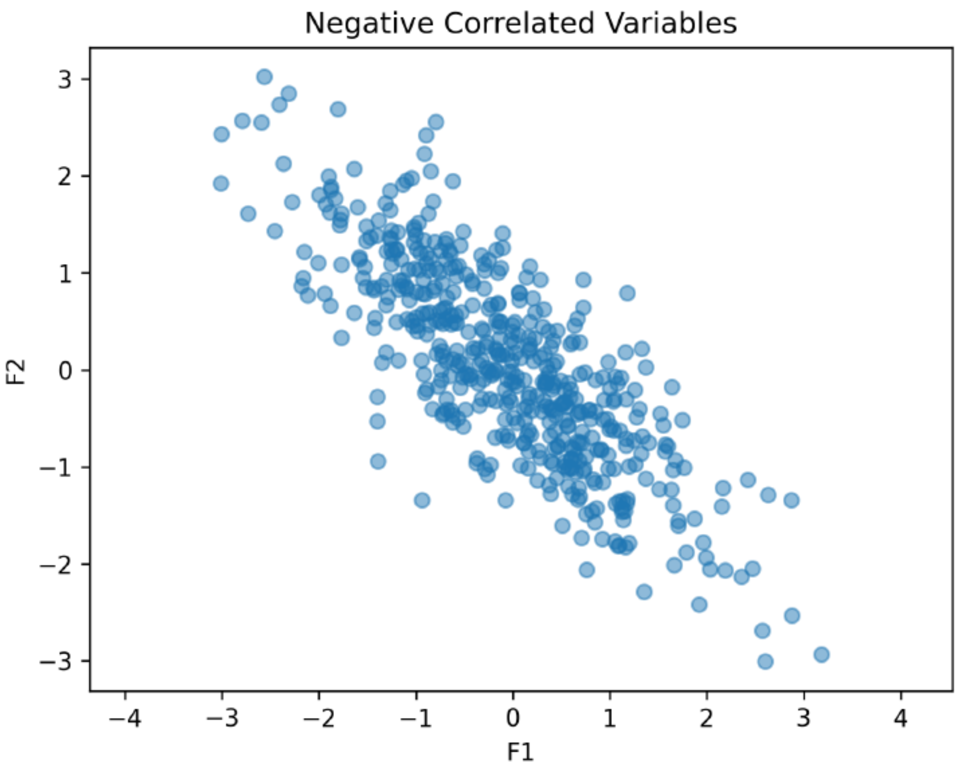
- If $\Sigma = \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix}$ and $a > d$, then:



- If $\Sigma = \begin{pmatrix} a & b \\ b & d \end{pmatrix}$, $a > d$, and $b > 0$, then:



- If $\Sigma = \begin{pmatrix} a & b \\ b & d \end{pmatrix}$, $a > d$, and $b < 0$, then:



Sample Covariance Matrix

- In practice, we estimate covariance from sample data.
- **Sample Covariance Matrix**: Given N samples of d features, the sample covariance matrix Σ is:

$$\Sigma_{d\times d} = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})(X_i - \bar{X})^T$$

- Where X_i is the i-th sample, and $\bar{X}_{d\times 1}$ is the mean of the samples.

Sample Covariance Example

- Suppose we have the following three samples each one having two features F_1 and F_2:

Sample	$\mathbf{F}_1$	$\mathbf{F}_2$
X_1	3	3
X_2	4	7
X_3	5	8
$\bar{X}$	4	6

$$\Sigma = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})(X_i - \bar{X})^T = \frac{1}{2}\left(\begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}\right) = \begin{pmatrix} 1 & 2.5 \\ 2.5 & 7 \end{pmatrix}$$

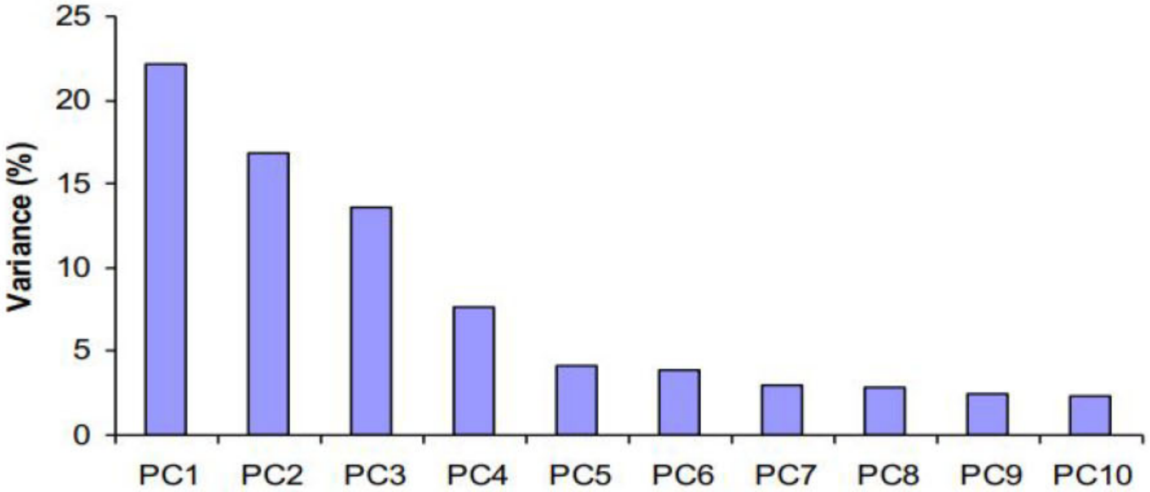
PCA

- Compute Eigenvalues and Eigenvectors of the Covariance matrix to Identify Principal Components.

- Why ?

Choose the number of PC

- For d original dimensions, the sample covariance matrix is $d \times d$, and has up to d eigenvectors. So we can have up to d principal components.
- Can ignore the components of lesser significance.



- We lose some information, but if the eigenvalues are small, we don't lose much.

Image compression

- Divide the original 372×492 image into patches.
    - Each patch is an instance containing 12×12 pixels on a grid.
- Consider each as a 144-D vector.

- 144D to 60D

- 144D to 16D

- 144D to 3D