Principal Component Analysis

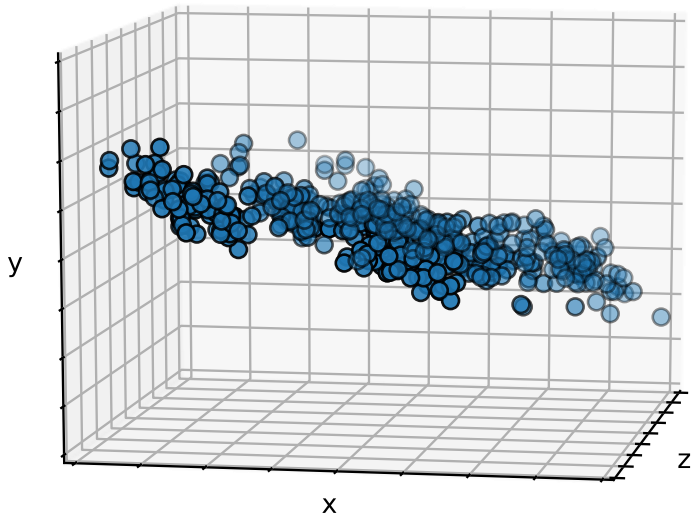
Finding the **best angle** to look at data from

Faculty of Mathematics and Computer Science, University of Bucharest
and
Sparktech Software

Academic Year 2018/2019, 1st Semester

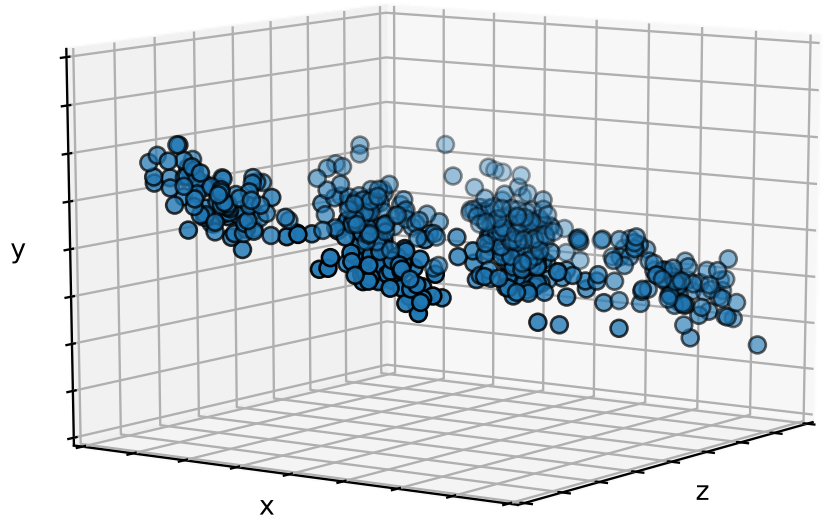
Intuition

- What can we do to get a *better view* of this dataset?



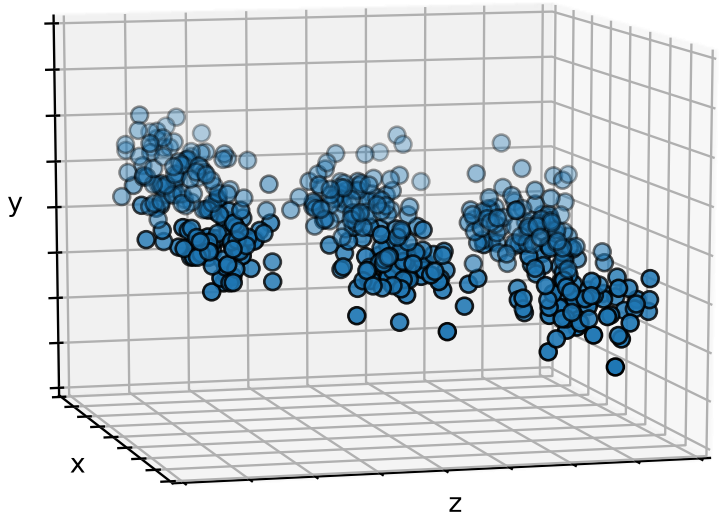
Intuition

- What can we do to get a *better view* of this dataset?
- We can try to rotate the axes until we find the *best angle* to look at the data from.



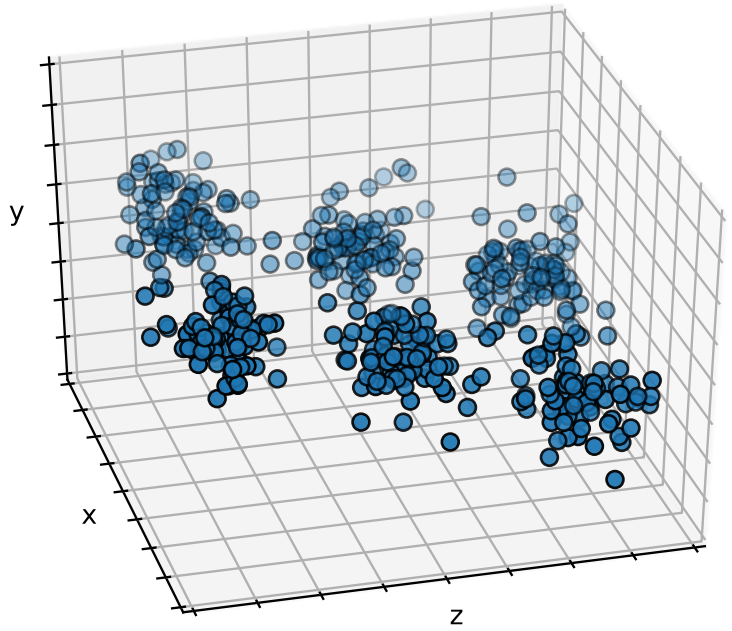
Intuition

- What can we do to get a *better view* of this dataset?
- We can try to rotate the axes until we find the *best angle* to look at the data from.



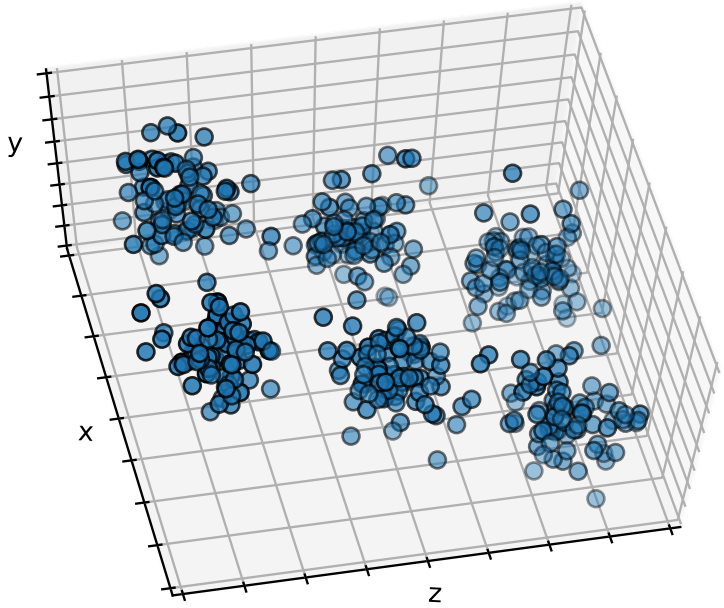
Intuition

- What can we do to get a *better view* of this dataset?
- We can try to rotate the axes until we find the *best angle* to look at the data from.



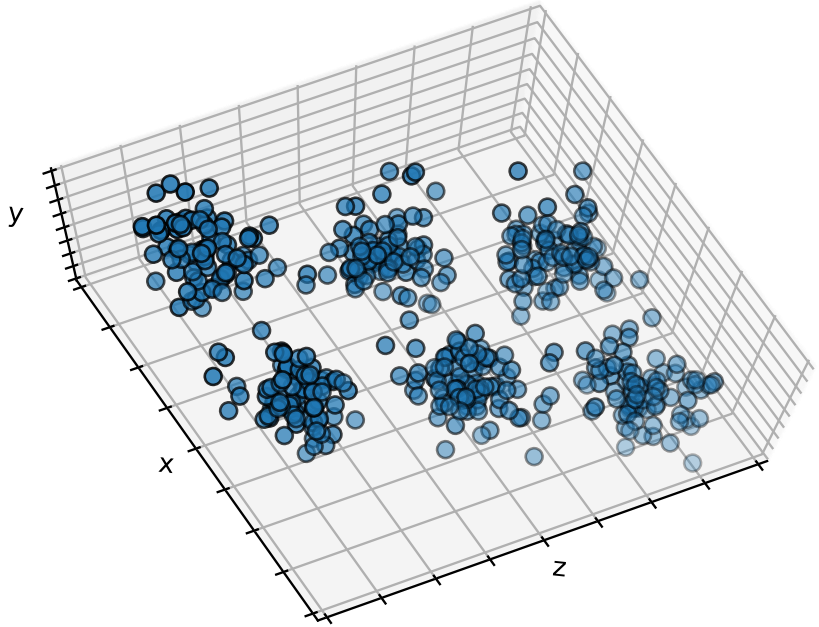
Intuition

- What can we do to get a *better view* of this dataset?
- We can try to rotate the axes until we find the *best angle* to look at the data from.



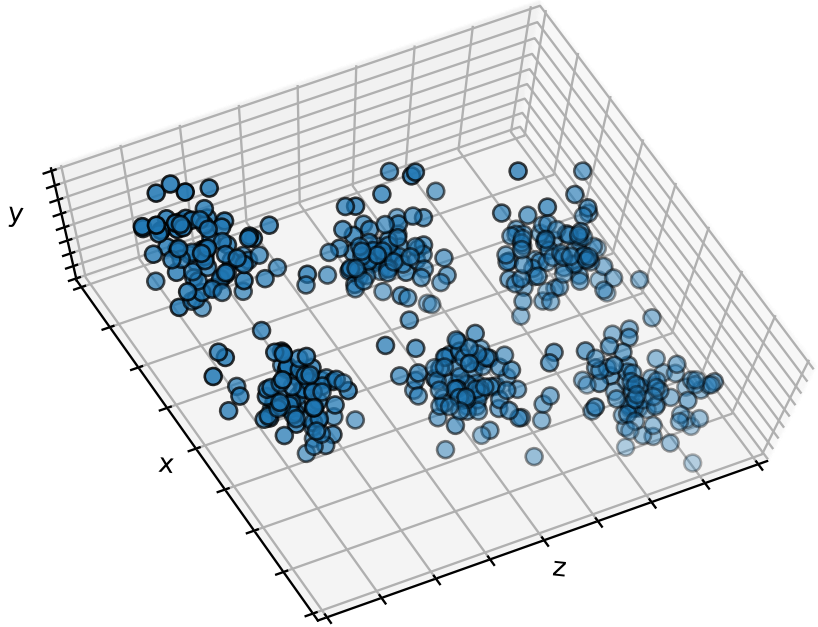
Intuition

- What can we do to get a *better view* of this dataset?
- We can try to rotate the axes until we find the *best angle* to look at the data from.



Intuition

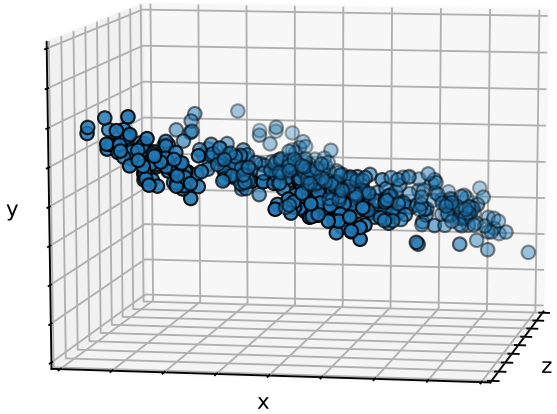
- What can we do to get a *better view* of this dataset?
- We can try to rotate the axes until we find the *best angle* to look at the data from.



We get a better understanding the data by looking from this angle than from our original perspective.

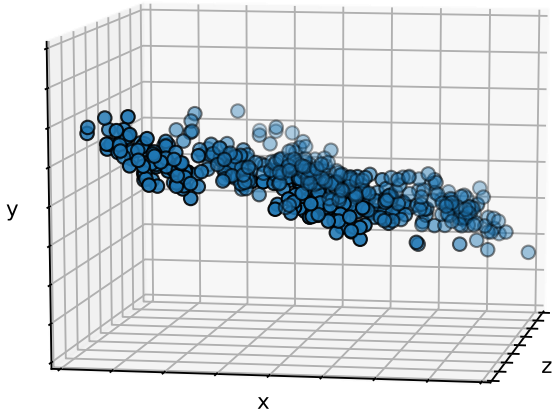
Intuition

- Rotation of the axes is equivalent to changing the *coordinate system* (or rotating the data).

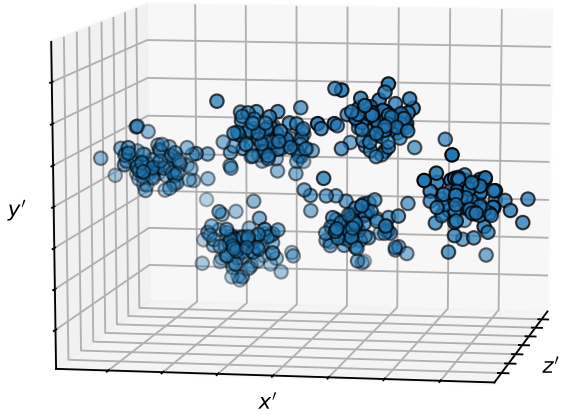


Intuition

- Rotation of the axes is equivalent to changing the *coordinate system* (or rotating the data).

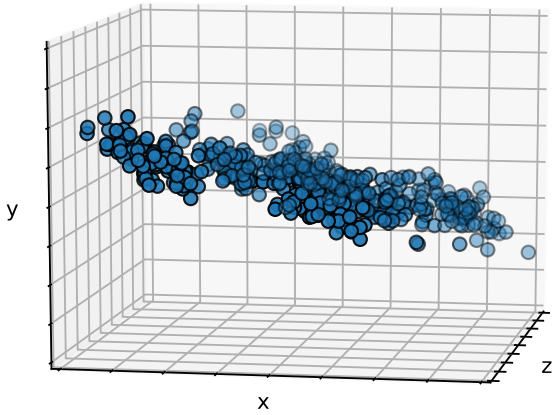


$$x' = 0.56x - 0.25y - 0.79z$$
$$y' = -0.75x + 0.25y - 0.61z$$
$$z' = 0.35x + 0.94y - 0.04z$$

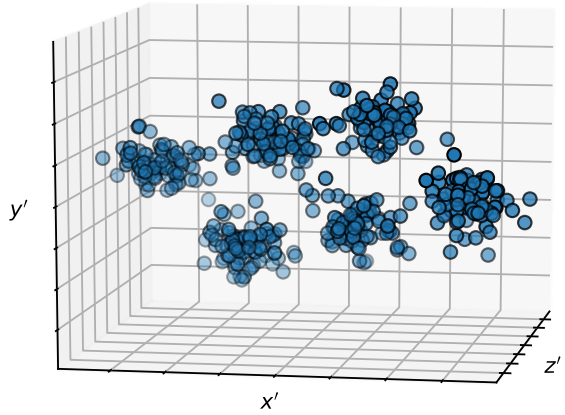


Intuition

- Rotation of the axes is equivalent to changing the *coordinate system* (or rotating the data).
- The axes which gives the *best view* of the data are called **principal components**.
  - They are the directions in which the dataset *varies* the most.

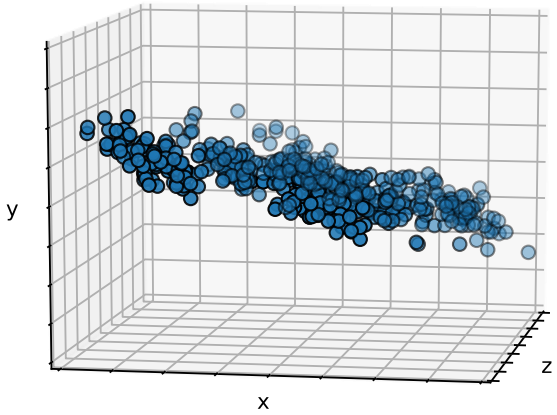


$$x' = 0.56x - 0.25y - 0.79z$$
$$y' = -0.75x + 0.25y - 0.61z$$
$$z' = 0.35x + 0.94y - 0.04z$$

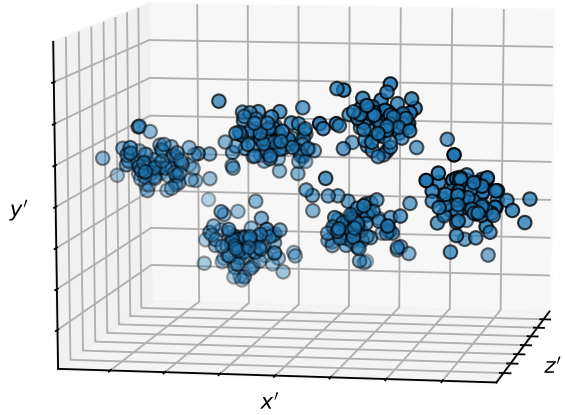


Intuition

- Rotation of the axes is equivalent to changing the *coordinate system* (or rotating the data).
- The axes which gives the *best view* of the data are called **principal components**.
  - They are the directions in which the dataset *varies* the most.



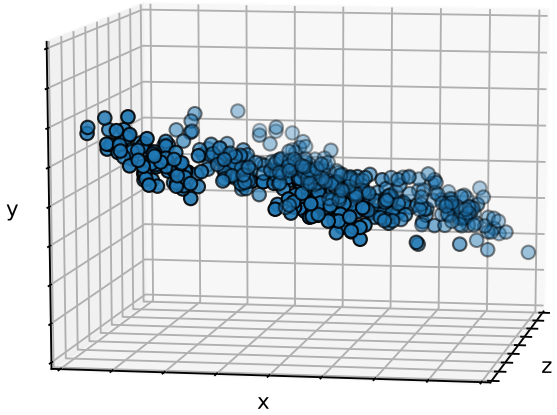
$$x' = 0.56x - 0.25y - 0.79z$$
$$y' = -0.75x + 0.25y - 0.61z$$
$$z' = 0.35x + 0.94y - 0.04z$$



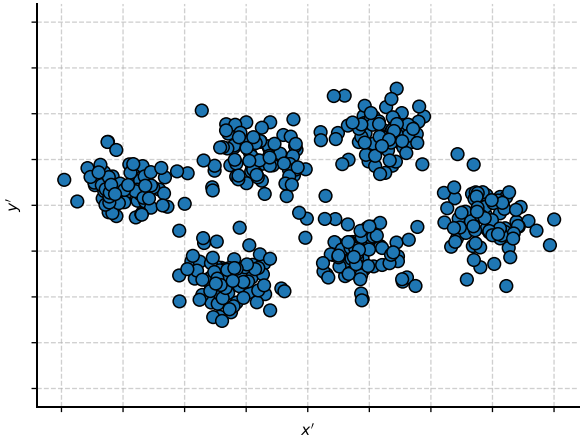
- We also compute *how much variance is explained* by each component
  - x' explains ~76%, y' explains ~26% and z' explains $< 1\%$

Intuition

- Rotation of the axes is equivalent to changing the *coordinate system* (or rotating the data).
- The axes which gives the *best view* of the data are called **principal components**.
  - They are the directions in which the dataset *varies* the most.



$$x' = 0.56x - 0.25y - 0.79z$$
$$y' = -0.75x + 0.25y - 0.61z$$
$$z' = 0.35x + 0.94y - 0.04z$$



- We also compute *how much variance is explained* by each component
  - x' explains ~76%, y' explains ~26% and z' explains $< 1\%$
  - We can discard components which explain too little variance.

Mathematical Preliminaries

Random Variable

- A **random variable** is a variable whose possible values are outcomes of a *random phenomenon* (e.g. rolling a die).
  - The source of uncertainty in a random variable can either be “objective” (the result of a *random process*) or “subjective” (the result of *incomplete knowledge*).
  - A random variable has a corresponding *probability distribution* which specifies the probability that its value falls in any given interval.

Random Variable

- A **random variable** is a variable whose possible values are outcomes of a *random phenomenon* (e.g. rolling a die).
  - The source of uncertainty in a random variable can either be “objective” (the result of a *random process*) or “subjective” (the result of *incomplete knowledge*).
  - A random variable has a corresponding *probability distribution* which specifies the probability that its value falls in any given interval.
- A **random variate** is a particular outcome of a *random variable*
  - The result of sampling its probability distribution.

Random Variable

- A **random variable** is a variable whose possible values are outcomes of a *random phenomenon* (e.g. rolling a die).
  - The source of uncertainty in a random variable can either be “objective” (the result of a *random process*) or “subjective” (the result of *incomplete knowledge*).
  - A random variable has a corresponding *probability distribution* which specifies the probability that its value falls in any given interval.
- A **random variate** is a particular outcome of a *random variable*
  - The result of sampling its probability distribution.
- The **expected value** E (or **mean** μ) of a *random variable* is the long-run average of its *random variates*.
  - For the discrete case, it is the *probability-weighted average* of all possible outcomes.

$$E[\text{🎲}] = 3.5$$

Random Variable

- A **random variable** is a variable whose possible values are outcomes of a *random phenomenon* (e.g. rolling a die).
  - The source of uncertainty in a random variable can either be “objective” (the result of a *random process*) or “subjective” (the result of *incomplete knowledge*).
  - A random variable has a corresponding *probability distribution* which specifies the probability that its value falls in any given interval.
- A **random variate** is a particular outcome of a *random variable*
  - The result of sampling its probability distribution.
- The **expected value** E (or **mean** μ) of a *random variable* is the long-run average of its *random variates*.
  - For the discrete case, it is the *probability-weighted average* of all possible outcomes.

$$E[\text{🎲}] = 3.5$$

- The **variance** σ^2 is a measure of the dispersion of *random variates* around the *expected value*.
  - σ is called the **standard deviation**.

Random Variable

- $\mathrm{X} = \{x_1, x_2, \dots, x_n\}$, $Y = \{y_1, y_2, \dots, y_n\}$ are two sets of random variates of two random variables which are sampled together (x_i at the same time with y_i).
- X and Y are just subsets of the whole (potentially infinite) population.

Random Variable

- $\mathrm{X} = \{x_1, x_2, \dots, x_n\},\ Y = \{y_1, y_2, \dots, y_n\}$ are two sets of random variates of two random variables which are sampled together (x_i at the same time with y_i).
- X and Y are just subsets of the whole (potentially infinite) population.
  - Then the **sample mean** and **sample variance** are just estimates of the true mean and variance values.

$$Mean(X) = E[X] = \mu_X = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$Var(X) = \sigma_X^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu_X)^2$$

Random Variable

- $\mathrm{X} = \{x_1, x_2, \dots, x_n\}$, $Y = \{y_1, y_2, \dots, y_n\}$ are two sets of random variates of two random variables which are sampled together (x_i at the same time with y_i).
- X and Y are just subsets of the whole (potentially infinite) population.
  - Then the **sample mean** and **sample variance** are just estimates of the true mean and variance values.

$$Mean(X) = E[X] = \mu_X = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$Var(X) = \sigma_X^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \mu_X)^2$$

$n-1$ instead of n is **Bessel's correction** which accounts for biased estimation of the mean.

Random Variable

- $\mathrm{X} = \{x_1, x_2, \dots, x_n\},\ Y = \{y_1, y_2, \dots, y_n\}$ are two sets of random variates of two random variables which are sampled together (x_i at the same time with y_i).
- X and Y are just subsets of the whole (potentially infinite) population.
  - Then the **sample mean** and **sample variance** are just estimates of the true mean and variance values.

$$Mean(X) = E[X] = \mu_X = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$Var(X) = \sigma_X^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \mu_X)^2$$

$n-1$ instead of n is **Bessel's correction** which accounts for biased estimation of the mean.

- **Covariance** measures the joint variability of two random variables (e.g. if larger values of one corresponds to larger values of the other, or vice-versa).

$$Cov(X, Y) = \sigma_{XY}^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \mu_X)(y_i - \mu_Y)$$

Features as Random Variables

- Dataset $X \in \mathbb{R}^{m \times n} = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}$ (m examples with n features).
  - Each row represents all features of an examples.
  - Each column represent the same feature of all examples.

Features as Random Variables

- Dataset $X \in \mathbb{R}^{m \times n} = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}$ (m examples with n features).
  - Each row represents all features of an examples.
  - Each column represent the same feature of all examples.
- We can regard each feature (each column) as a *random variable*.
  - This makes the value $x_j^{(i)}$ (feature j of example i) a *random variate*.
  - Example $\vec{x}^{(i)}$ is made up of n *random variables* sampled together.

Features as Random Variables

- Dataset $X \in \mathbb{R}^{m \times n} = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}$ (m examples with n features).
  - Each row represents all features of an examples.
  - Each column represent the same feature of all examples.
- We can regard each feature (each column) as a *random variable*.
  - This makes the value $x_j^{(i)}$ (feature j of example i) a *random variate*.
  - Example $\vec{x}^{(i)}$ is made up of n *random variables* sampled together.
- This makes dataset X a set of m sampling events of n random variables.
  - $\Longrightarrow$ We can compute *mean*, *variance*, *covariance*.

Features as Random Variables

$$X = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}_{m \times n}$$

$$\mu_X = (\mu_1 \quad \mu_2 \quad \cdots \quad \mu_n) = \frac{1}{m}\left(\sum_{i=1}^{m} x_1^{(i)} \quad \sum_{i=1}^{m} x_2^{(i)} \quad \cdots \quad \sum_{i=1}^{m} x_n^{(i)}\right)$$

$$\bar{X} = \begin{pmatrix} x_1^{(1)} - \mu_1 & \cdots & x_n^{(1)} - \mu_n \\ x_1^{(2)} - \mu_1 & \cdots & x_n^{(2)} - \mu_n \\ \vdots & \ddots & \vdots \\ x_1^{(m)} - \mu_1 & \cdots & x_n^{(m)} - \mu_n \end{pmatrix} = X - 1_{m \times 1} \cdot \mu_X$$

Centered version of X

Features as Random Variables

$$\bar{X} = \begin{pmatrix} x_1^{(1)} - \mu_1 & x_1^{(1)} - \mu_1 & \cdots & x_n^{(1)} - \mu_n \\ x_1^{(2)} - \mu_1 & x_1^{(1)} - \mu_1 & \cdots & x_n^{(2)} - \mu_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} - \mu_1 & x_1^{(1)} - \mu_1 & \cdots & x_n^{(m)} - \mu_n \end{pmatrix}$$

$$\bar{X}^T = \begin{pmatrix} x_1^{(1)} - \mu_1 & x_1^{(2)} - \mu_1 & \cdots & x_1^{(m)} - \mu_1 \\ x_2^{(1)} - \mu_2 & x_2^{(2)} - \mu_2 & \cdots & x_2^{(m)} - \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_n^{(1)} - \mu_n & x_n^{(2)} - \mu_n & \cdots & x_n^{(m)} - \mu_n \end{pmatrix}$$

$$\frac{1}{m-1}\bar{X}^T\bar{X} =$$

Features as Random Variables

$$\bar{X} = \begin{pmatrix} x_1^{(1)} - \mu_1 & x_1^{(1)} - \mu_1 & \cdots & x_n^{(1)} - \mu_n \\ x_1^{(2)} - \mu_1 & x_1^{(1)} - \mu_1 & \cdots & x_n^{(2)} - \mu_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} - \mu_1 & x_1^{(1)} - \mu_1 & \cdots & x_n^{(m)} - \mu_n \end{pmatrix} \qquad \bar{X}^T = \begin{pmatrix} x_1^{(1)} - \mu_1 & x_1^{(2)} - \mu_1 & \cdots & x_1^{(m)} - \mu_1 \\ x_2^{(1)} - \mu_2 & x_2^{(2)} - \mu_2 & \cdots & x_2^{(m)} - \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_n^{(1)} - \mu_n & x_n^{(2)} - \mu_n & \cdots & x_n^{(m)} - \mu_n \end{pmatrix}$$

$$\frac{1}{m-1}\bar{X}^T\bar{X} = \frac{1}{m-1}\begin{pmatrix} \sum_{i=1}^{n}\left(x_1^{(i)} - \mu_1\right)^2 & \sum_{i=1}^{n}\left(x_1^{(i)} - \mu_1\right)\left(x_2^{(i)} - \mu_2\right) & \cdots & \sum_{i=1}^{n}\left(x_1^{(i)} - \mu_1\right)\left(x_n^{(i)} - \mu_n\right) \\ \sum_{i=1}^{n}\left(x_2^{(i)} - \mu_2\right)\left(x_1^{(i)} - \mu_1\right) & \sum_{i=1}^{n}\left(x_2^{(i)} - \mu_2\right)^2 & \cdots & \sum_{i=1}^{n}\left(x_2^{(i)} - \mu_2\right)\left(x_n^{(i)} - \mu_n\right) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n}\left(x_n^{(i)} - \mu_n\right)\left(x_1^{(i)} - \mu_1\right) & \sum_{i=1}^{n}\left(x_n^{(i)} - \mu_n\right)\left(x_2^{(i)} - \mu_2\right) & \cdots & \sum_{i=1}^{n}\left(x_n^{(i)} - \mu_n\right)^2 \end{pmatrix}$$

Features as Random Variables

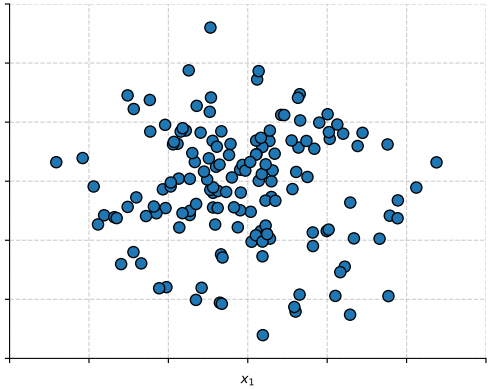
$$\bar{X} = \begin{pmatrix} x_1^{(1)} - \mu_1 & x_1^{(1)} - \mu_1 & \cdots & x_n^{(1)} - \mu_n \\ x_1^{(2)} - \mu_1 & x_1^{(1)} - \mu_1 & \cdots & x_n^{(2)} - \mu_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} - \mu_1 & x_1^{(1)} - \mu_1 & \cdots & x_n^{(m)} - \mu_n \end{pmatrix} \qquad \bar{X}^T = \begin{pmatrix} x_1^{(1)} - \mu_1 & x_1^{(2)} - \mu_1 & \cdots & x_1^{(m)} - \mu_1 \\ x_2^{(1)} - \mu_2 & x_2^{(2)} - \mu_2 & \cdots & x_2^{(m)} - \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_n^{(1)} - \mu_n & x_n^{(2)} - \mu_n & \cdots & x_n^{(m)} - \mu_n \end{pmatrix}$$

$$\frac{1}{m-1}\bar{X}^T\bar{X} = \frac{1}{m-1}\begin{pmatrix} \sum_{i=1}^{n}\left(x_1^{(i)} - \mu_1\right)^2 & \sum_{i=1}^{n}\left(x_1^{(i)} - \mu_1\right)\left(x_2^{(i)} - \mu_2\right) & \cdots & \sum_{i=1}^{n}\left(x_1^{(i)} - \mu_1\right)\left(x_n^{(i)} - \mu_n\right) \\ \sum_{i=1}^{n}\left(x_2^{(i)} - \mu_2\right)\left(x_1^{(i)} - \mu_1\right) & \sum_{i=1}^{n}\left(x_2^{(i)} - \mu_2\right)^2 & \cdots & \sum_{i=1}^{n}\left(x_2^{(i)} - \mu_2\right)\left(x_n^{(i)} - \mu_n\right) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n}\left(x_n^{(i)} - \mu_n\right)\left(x_1^{(i)} - \mu_1\right) & \sum_{i=1}^{n}\left(x_n^{(i)} - \mu_n\right)\left(x_2^{(i)} - \mu_2\right) & \cdots & \sum_{i=1}^{n}\left(x_n^{(i)} - \mu_n\right)^2 \end{pmatrix}$$

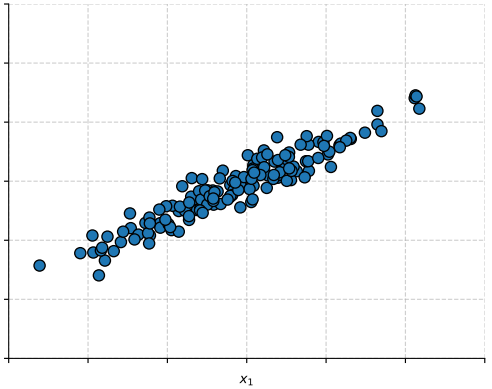
$$= \begin{pmatrix} Var(x_1) & Cov(x_1, x_2) & \cdots & Cov(x_1, x_n) \\ Cov(x_1, x_2) & Var(x_2) & \cdots & Cov(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(x_1, x_n) & Cov(x_2, x_n) & \cdots & Var(x_n) \end{pmatrix} \stackrel{\text{def}}{=} Cov(X) = \Sigma_X$$

Covariance of a dataset

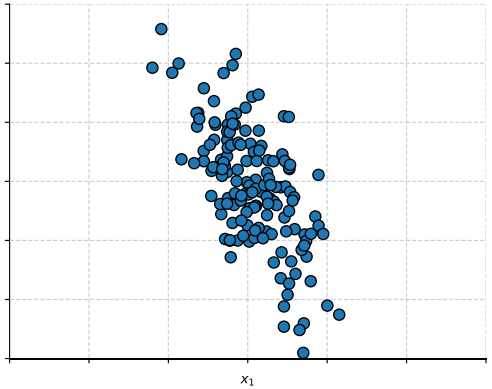
$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



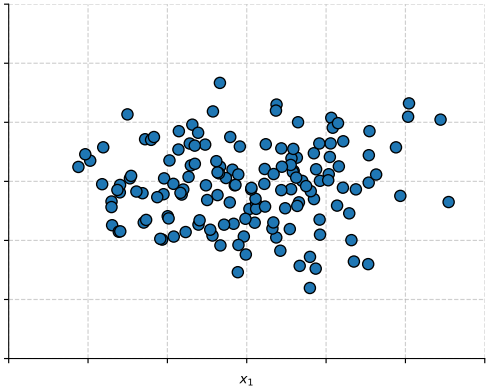
$$\Sigma = \begin{pmatrix} 1 & 0.6 \\ 0.6 & 0.4 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 0.2 & -0.3 \\ -0.3 & 1 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$$



Eigenvectors and eigenvalues

- An **eigenvector** $v \in \mathbb{R}^n$ and its corresponding **eigenvalue** $\lambda \in \mathbb{R}$ of a square matrix $A \in \mathbb{R}^{n \times n}$ are the solutions to the following equation:

$$Av = \lambda v$$

- In other words, they are the vectors which *only get scaled* when multiplied by A, but *don't change direction*. The corresponding eigenvalue λ is the scaling factor.

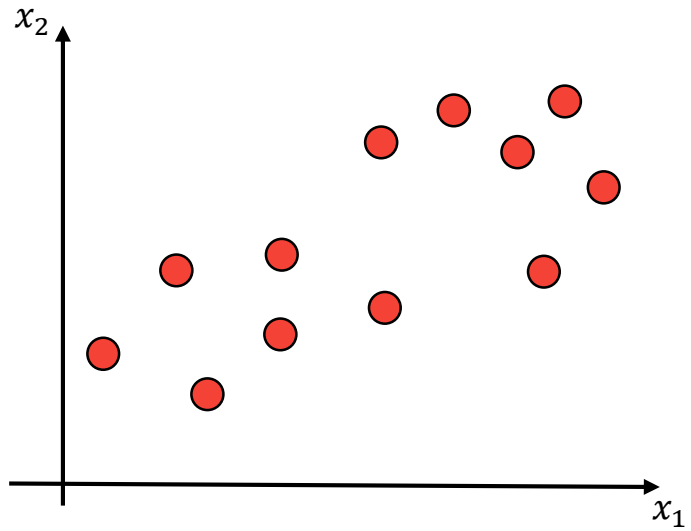
Finding Principal Components

Principal Component Analysis

- **Principal Component Analysis** (**PCA**) transforms a dataset into a new *orthogonal coordinate system* in which the data is *centered* and the features are completely *uncorrelated*.
  - The *mean* of the new dataset is 0.
  - The *covariance* of any pair of distinct features is 0.
- The features of the transformed dataset (called **principal components**) are sorted by *variance*, such that the first feature has the greatest variance.
- Component with low variance can be discarded, making *PCA* a method of **dimensionality reduction**.

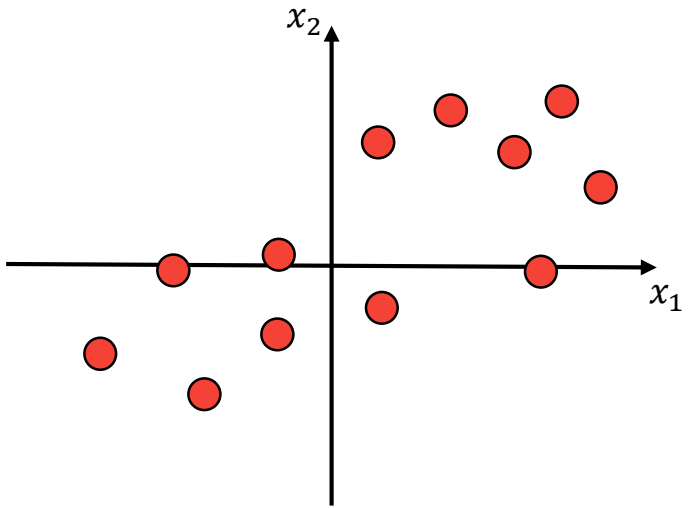
Finding the first PC

- The first step is *centering* the data (subtracting the *mean* from all data points)..



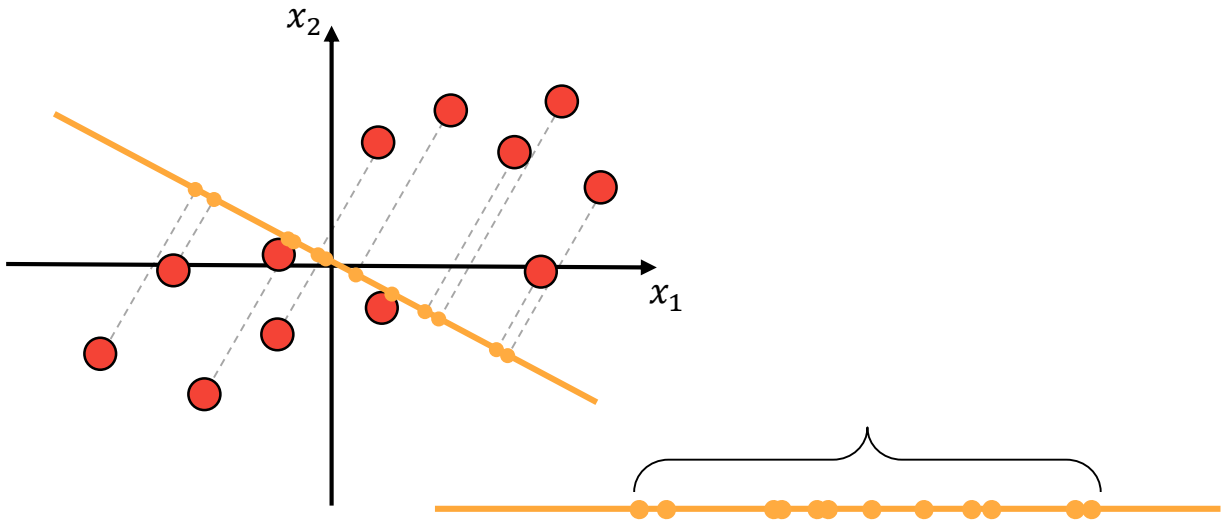
Finding the first PC

- The first step is *centering* the data (subtracting the *mean* from all data points).



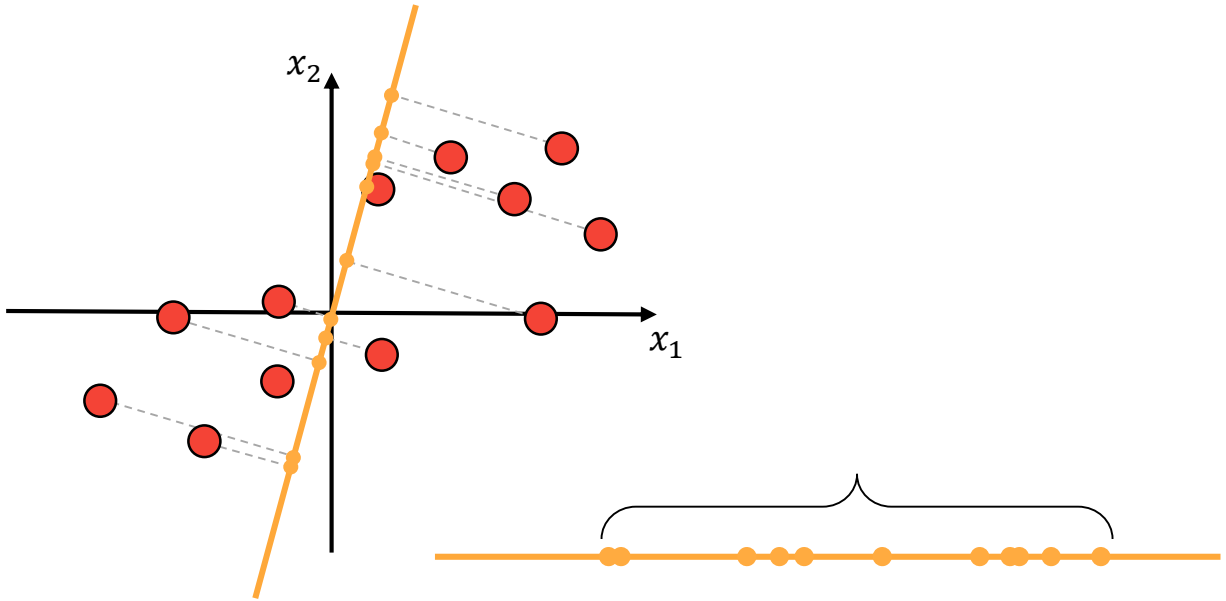
Finding the first PC

- The first step is *centering* the data (subtracting the *mean* from all data points).
- The first *principal component* is the direction on which the data has the *largest variance*.
  - We are looking for a line on which the projection of data points are as spread out as possible.



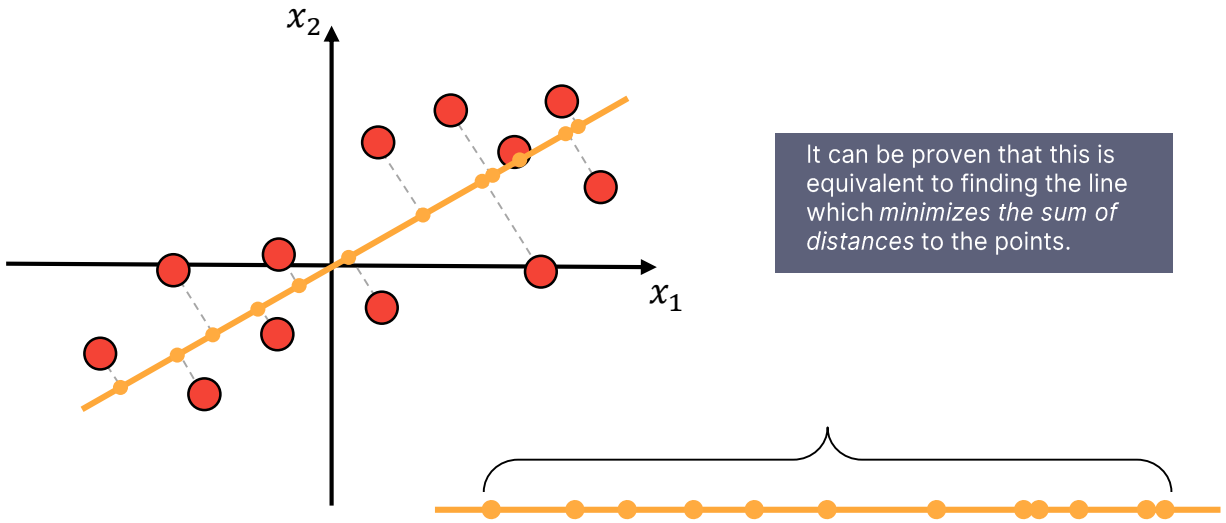
Finding the first PC

- The first step is *centering* the data (subtracting the *mean* from all data points).
- The first *principal component* is the direction on which the data has the *largest variance*.
  - We are looking for a line on which the projection of data points are as spread out as possible.



Finding the first PC

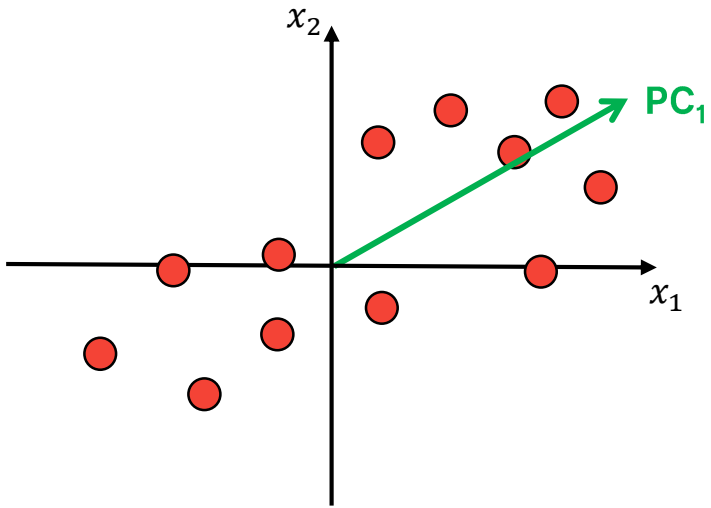
- The first step is *centering* the data (subtracting the *mean* from all data points).
- The first *principal component* is the direction on which the data has the *largest variance*.
  - We are looking for a line on which the projection of data points are as spread out as possible.



It can be proven that this is equivalent to finding the line which *minimizes the sum of distances* to the points.

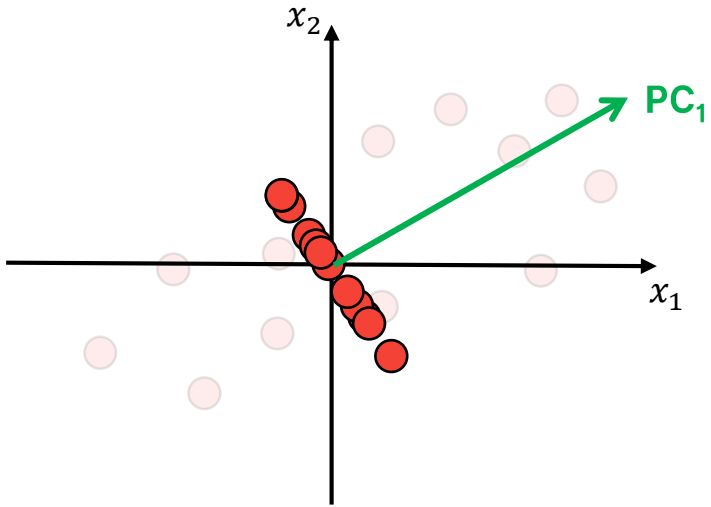
Finding further PCs

- If we subtract the projection of points on the first PC from the points themselves, we get a dataset which has 0 variance on that direction.



Finding further PCs

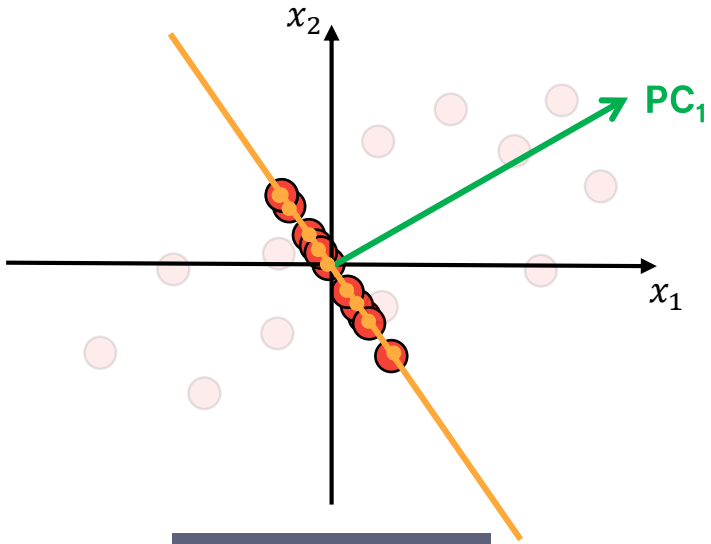
- If we subtract the projection of points on the first PC from the points themselves, we get a dataset which has 0 variance on that direction.



Finding further PCs

- If we subtract the projection of points on the first PC from the points themselves, we get a dataset which has 0 variance on that direction.
- By applying the same method on the new dataset, we get the *second principal component* of the original dataset.

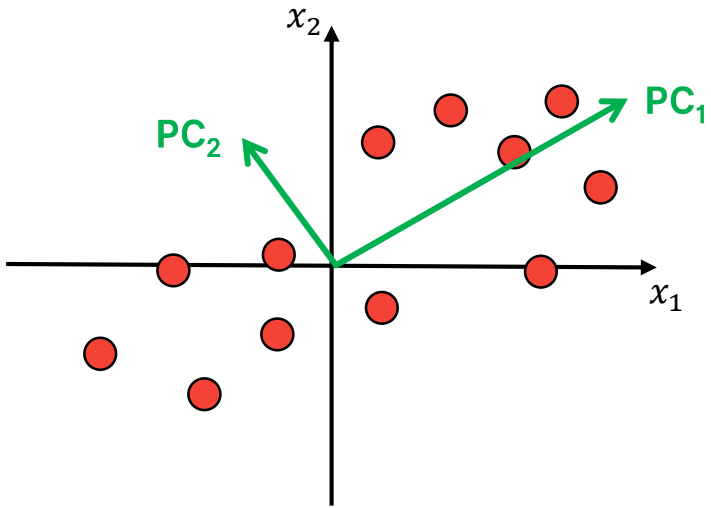
The second PC is the direction in which the data varies the most, after eliminating the variance on the first PC.



Trivial in 2 dimensions.

Finding further PCs

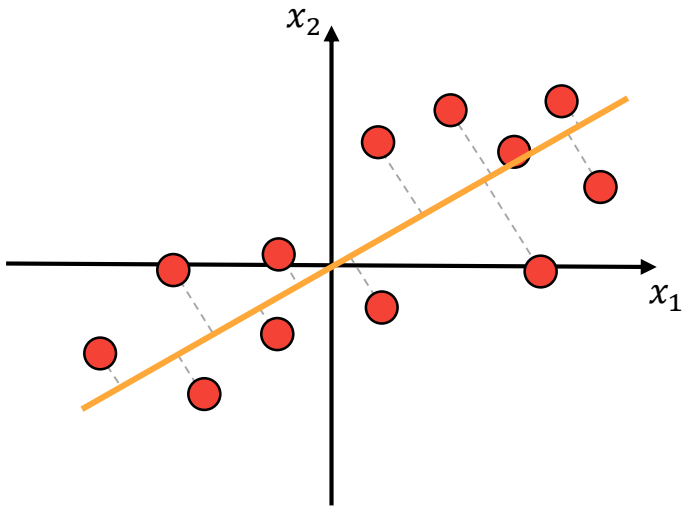
- If we subtract the projection of points on the first PC from the points themselves, we get a dataset which has 0 variance on that direction.
- By applying the same method on the new dataset, we get the *second principal component* of the original dataset.



If we had more than 2 dimensions, the process can continue.

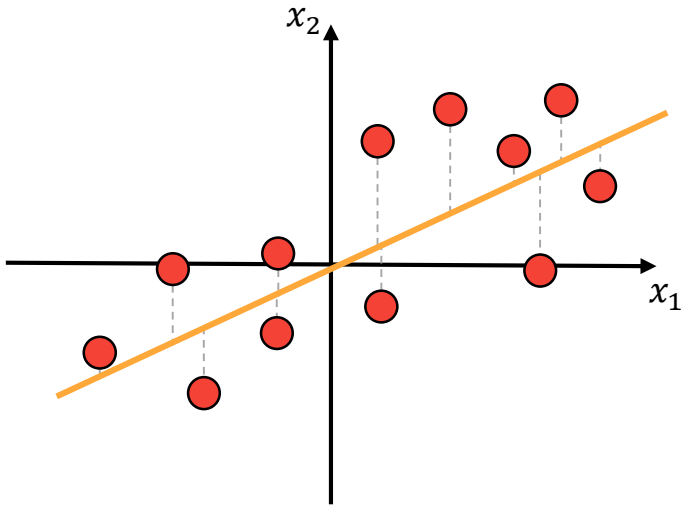
PCA is not Linear Regression!

PCA finds the line which minimizes the *sum of distances to the data points.*



x_1 and x_2 are both features
(a.k.a. independent variable)

Linear Regression finds the line which minimizes the *sum of squared distances to the predictions* given by the line.

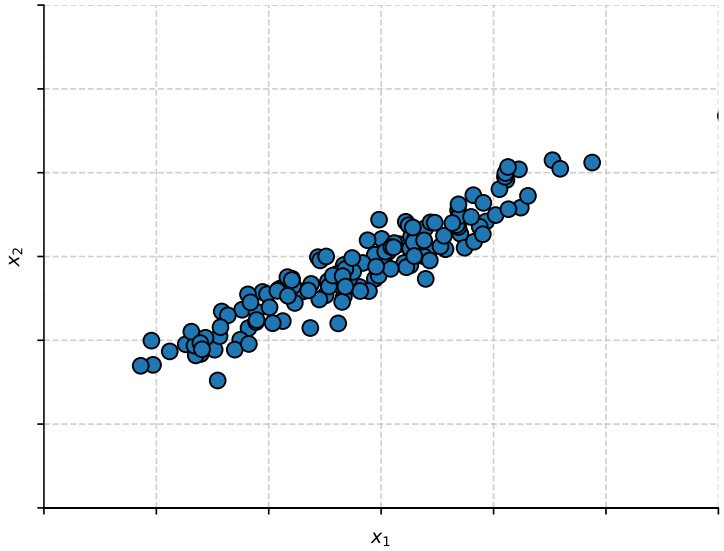


x_2 is a label
(a.k.a. dependent variable)

PCA as
Eigen Decomposition

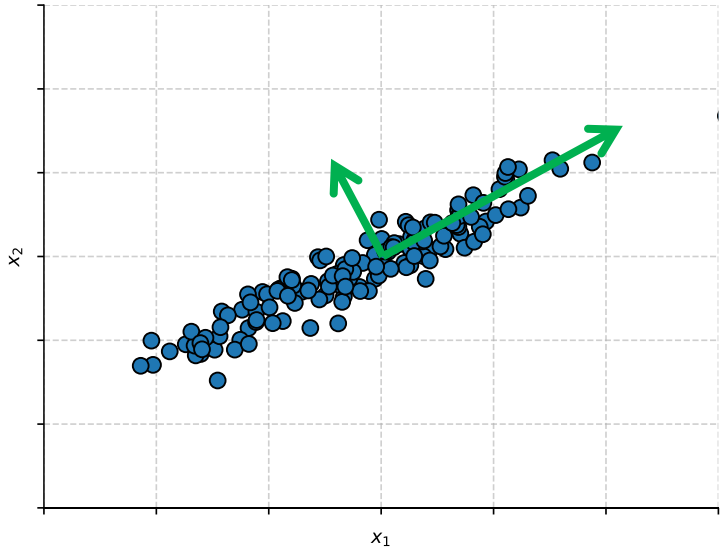
PCA as Eigen Decomposition

- $X \in \mathbb{R}^{m \times n}$ with covariance Σ_X.
- How do the eigenvectors of Σ_X look like?
  - Intuitively, consider Σ_X "responsible" for X's shape.
  - Remember than the *eigenvectors* are the vectors which do not change direction when multiplied by Σ_X.



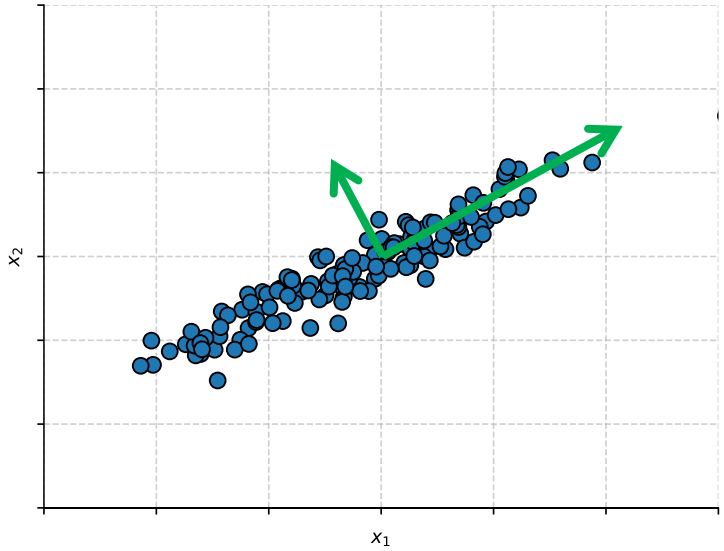
PCA as Eigen Decomposition

- $X \in \mathbb{R}^{m \times n}$ with covariance Σ_X.
- How do the eigenvectors of Σ_X look like?
  - Intuitively, consider Σ_X "responsible" for X's shape.
  - Remember than the *eigenvectors* are the vectors which do not change direction when multiplied by Σ_X.
- The **eigenvectors** of the *covariance matrix* are, in fact, the **principal components** of matrix X.
  - The corresponding **eigenvalues** are equal to the *amount of explained variance* of each component.



PCA as Eigen Decomposition

- $X \in \mathbb{R}^{m \times n}$ with covariance Σ_X.
- How do the eigenvectors of Σ_X look like?
  - Intuitively, consider Σ_X "responsible" for X's shape.
  - Remember than the *eigenvectors* are the vectors which do not change direction when multiplied by Σ_X.
- The **eigenvectors** of the *covariance matrix* are, in fact, the **principal components** of matrix X.
  - The corresponding **eigenvalues** are equal to the *amount of explained variance* of each component.



- Eigen Decomposition:

$$\Sigma_X = V \Lambda V^{-1}$$

where

$$V \in \mathbb{R}^{n \times n} = (v_1 \quad \cdots \quad v_n), \quad v_i \in \mathbb{R}^{n \times 1} \text{ - column eigenvectors}$$

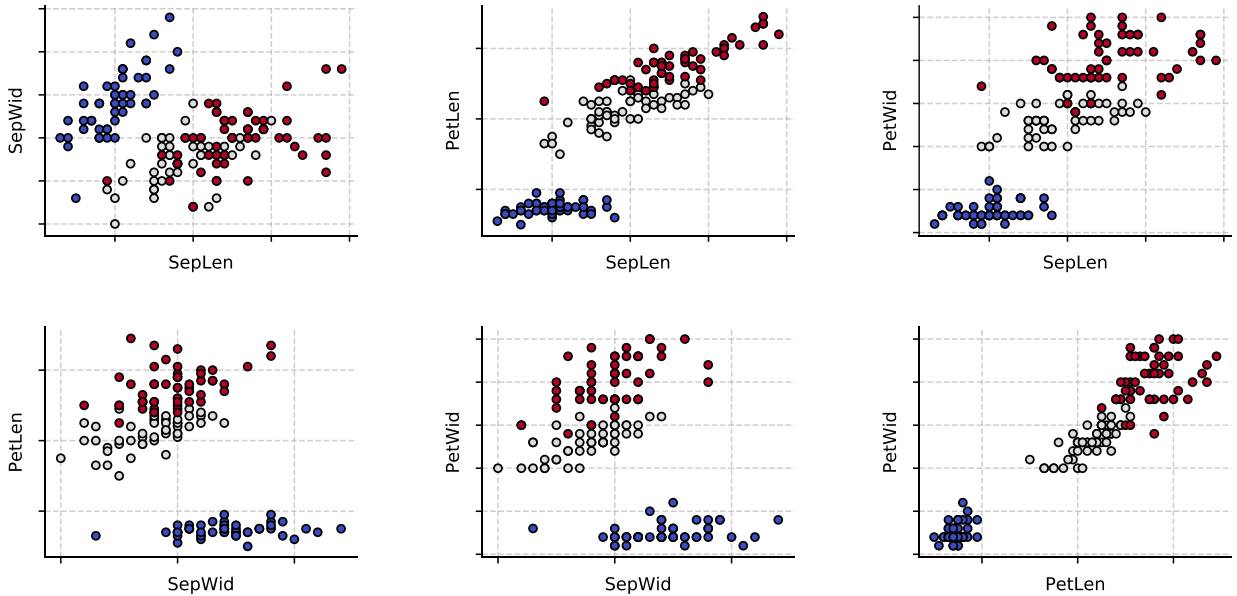
$$\Lambda \in \mathbb{R}^{n \times n} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} \text{ - diagonal eigenvalues}$$

PCA as Eigen Decomposition

```
1  def PCA(X):
2      X̄ = X − μ_X  # center X
3      Σ_X = 1/(m−1) X̄^T X̄  # compute covariance
4      V, Λ = eig_decompose(Σ_X)  # Σ_X = VΛV^{−1} eigen decomposition of covariance
5      V′ = V[argsort_reversed(diag(Λ))]  # permute columns by decreasing λ_i
6      k = number_of_pcs_to_keep(Λ)  # optionally, keep only k PCs which explain most variance.
7      V′ = V′[:, :k]  # V′ ∈ ℝ^{n×k}
8      X′ = XV′  # X′ ∈ ℝ^{m×k} (if we want the same dimensionality as X, we can pad with zeros)
9      return X′
```

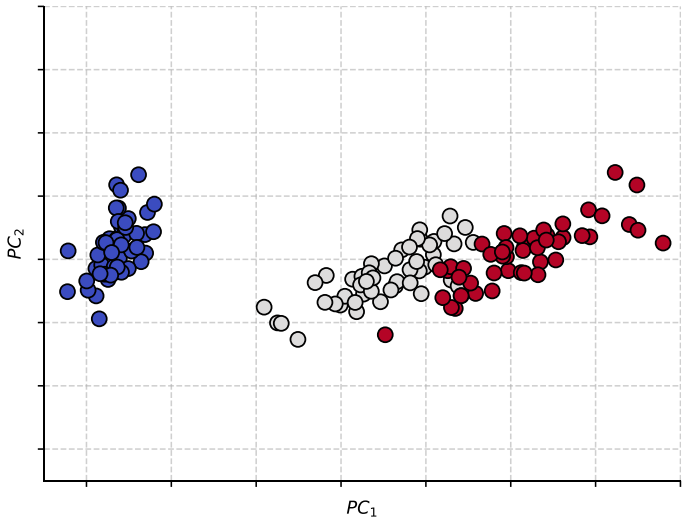
Example

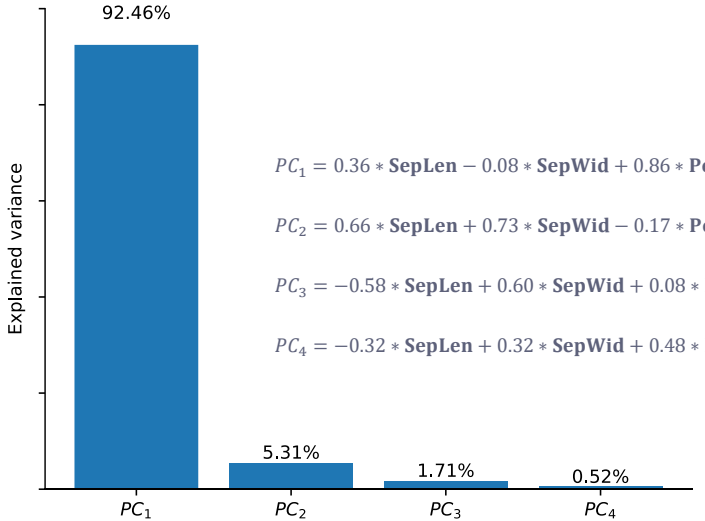
- Iris Dataset – 150 samples of flowers with 4 features and 3 classes.
  - $X \in \mathbb{R}^{150\times4}$
  - Features: *Sepal Length* (cm), *Sepal Width* (cm), *Petal Length* (cm), *Petal Width* (cm)



Example

- Iris Dataset – 150 samples of flowers with 4 features and 3 classes.
  - $X \in \mathbb{R}^{150\times4}$
  - Features: *Sepal Length* (cm), *Sepal Width* (cm), *Petal Length* (cm), *Petal Width* (cm)





$$PC_1 = 0.36 * \mathbf{SepLen} - 0.08 * \mathbf{SepWid} + 0.86 * \mathbf{PetLen} + 0.36 * \mathbf{PetWid}$$

$$PC_2 = 0.66 * \mathbf{SepLen} + 0.73 * \mathbf{SepWid} - 0.17 * \mathbf{PetLen} - 0.08 * \mathbf{PetWid}$$

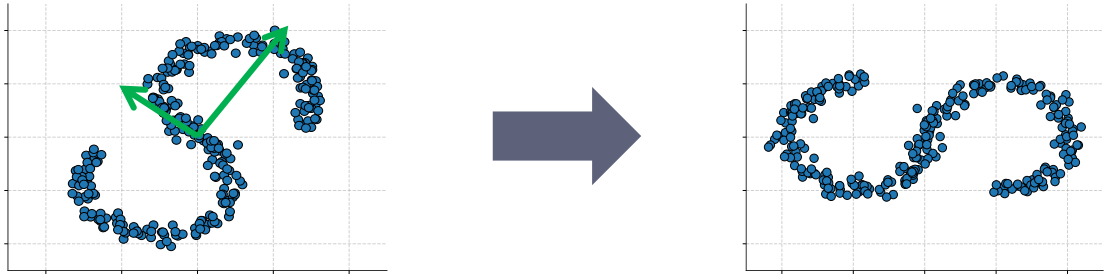
$$PC_3 = -0.58 * \mathbf{SepLen} + 0.60 * \mathbf{SepWid} + 0.08 * \mathbf{PetLen} + 0.55 * \mathbf{PetWid}$$

$$PC_4 = -0.32 * \mathbf{SepLen} + 0.32 * \mathbf{SepWid} + 0.48 * \mathbf{PetLen} - 0.75 * \mathbf{PetWid}$$

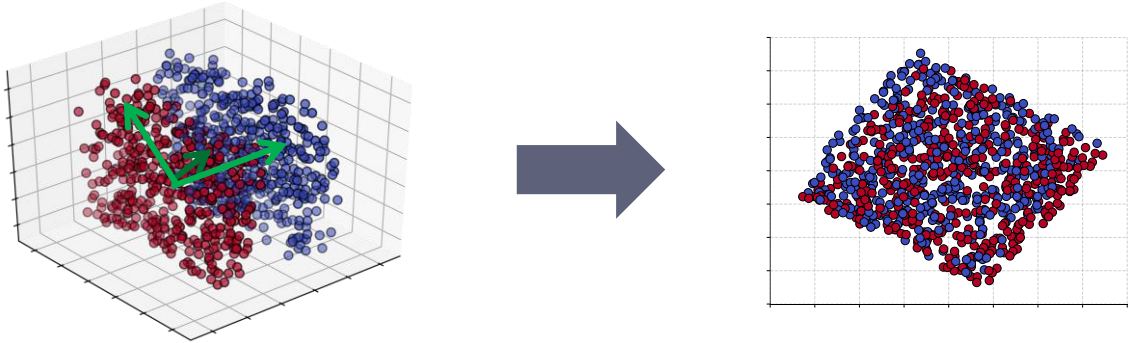
First 2 components explain almost 98% of variance.

Limitations

- PCA does not consider non-linear correlation:



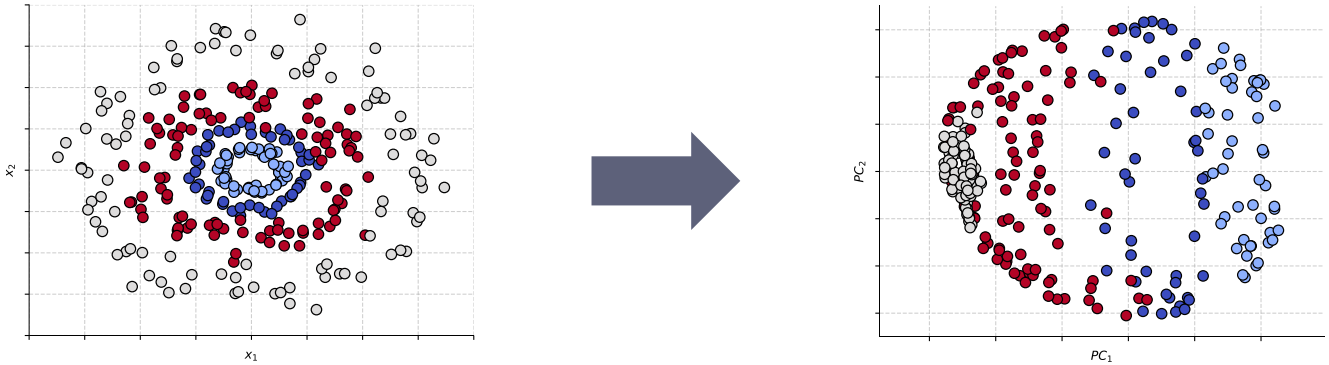
- PCA does not take label into account, only variance in features:



Beyond PCA

Kernel PCA

- $\phi: \mathbb{R}^n \to \mathbb{R}^N, N > n$ is a mapping to a higher-dimensional feature space.
  - $\phi(X) \in \mathbb{R}^{m \times N} = \begin{pmatrix} & \cdots & \\ \vdots & \phi\left(x_j^{(i)}\right) & \vdots \\ & \cdots & \end{pmatrix}$ is the mapping of the whole dataset X.
- **Kernel PCA** is a method of doing *PCA* on $\phi(X)$ without explicitly computing (or even knowing) the mapping ϕ, by using the *kernel trick*.
  - We are only computing the projection of the points on the PCs of $\phi(X)$, not the components themselves.



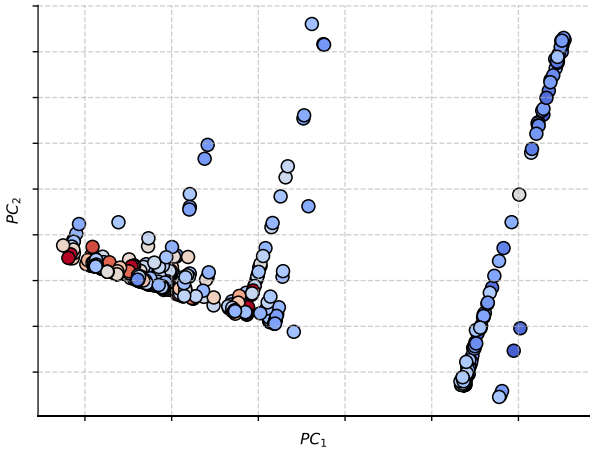
Partial Least Squares Regression

- **PLS Regression** is a *multivariate* regression method which bears some similarity to PCA, in the sense that it finds certain directions in feature space.
- Unlike PCA, which finds directions with the most variance in *feature space*, PLS finds the directions in *feature space* which explain most of the variance in *label space*.
    - PLS is a supervised technique.

Partial Least Squares Regression – Example

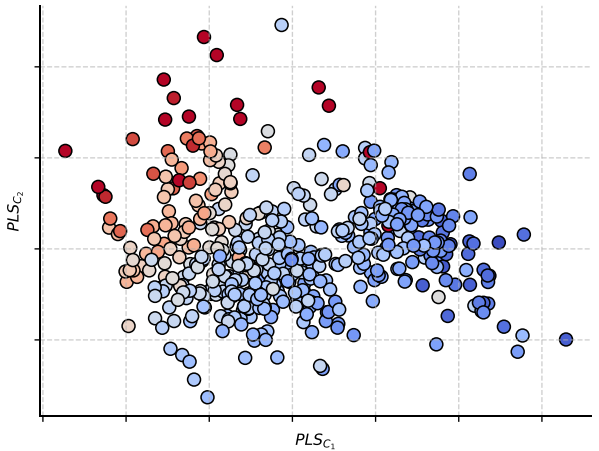
- Boston House Prices Dataset – 506 samples with 13 features and 1 target.
  - Features: *Crime Rate per Capita*, *Nitric Oxides Concentration*, *Index of Accessibility to Highways*, *Pupil-Teacher Ratio by Town*, etc.
  - Target: *Median value of Homes* (in $1000s)

PCA



Target variable is coded by color.

First 2 directions of PLS

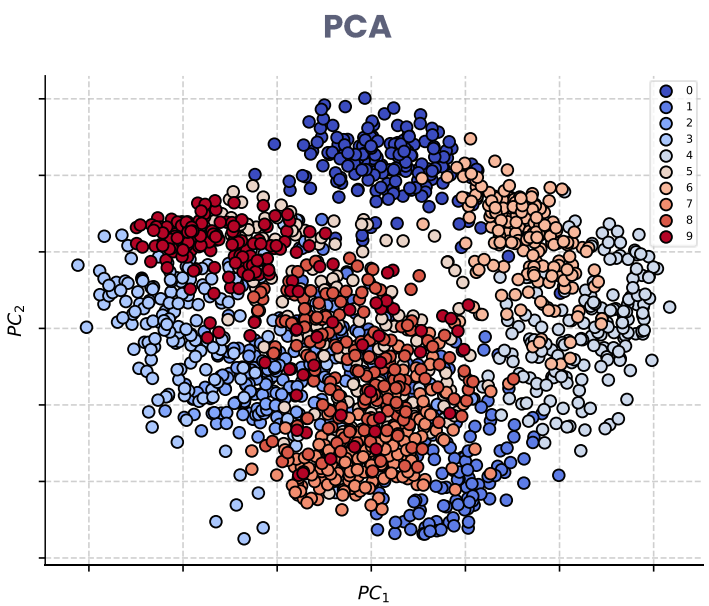


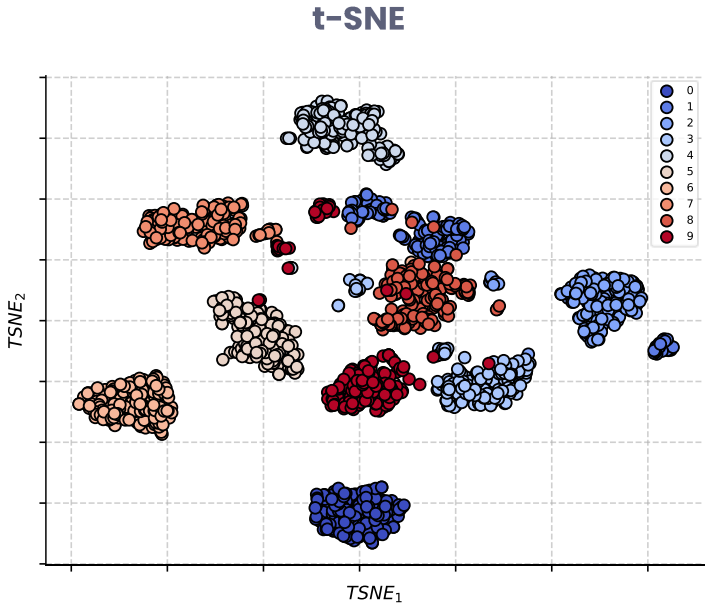
t-SNE

- **t-Distributed Stochastic Neighbor Embedding (t-SNE)** is a *non-linear dimensionality reduction* technique.
- It maps the data points in a lower-dimensional, space such that points which were close in the original space remain close in the embedding-space.
  - It focuses on preserving local structure and does not preserve global structure.
  - The results are not deterministic.
  - It is computationally expensive
- t-SNE does not produce a transformation of the space, it generates an *embedding* into a completely new space.
  - The features in the new space are difficult to interpret.
  - It is mostly used as a tool for visualization.

t-SNE – Example

- *UCI Optical Recognition of Handwritten Digits* Dataset - 5620 samples of 8x8 pixel images with handwritten digits (9 classes)





PCA in Python

```
from sklearn.decomposition import PCA, KernelPCA
from sklearn.cross_decomposition import PLSRegression
from sklearn.manifold import TSNE

pca = PCA(n_components = 2) # number of components to keep, if “None” it keeps all components
pca.fit(X) # find the principal components
X_pca = pca.transform(X) # rotates X into the new coordinate system
pca.components_ # coeficients of the original components to produce the new components
pca.explained_variance_ # variance of projections per component
pca.explained_variance_ratio_ # ratio of explained variance per component
```

Summary

- **Principal Component Analysis** (**PCA**) transforms a dataset into a new *orthogonal coordinate system* in which the data is *centered* and the features are completely *uncorrelated*.
- The directions of the new coordinate system are called **principal components.**
  - They are sorted by *variance*, such that the first PC has the greatest variance.
- Component with low variance can be discarded, making *PCA* a method of **dimensionality reduction**.
- PCA does not treat *non-linear correlation* and does not take label into account.

Keywords

Principle Component Analysis | PCA

Random Variable | Mean | Variance | Covariance | Bessel's Correction

Covariance Matrix | Eigenvector | Eigenvalue

Dimensionality Reduction

Kernel PCA | PLS Regression | t-SNE