Machine learning algorithms: Practice 4.

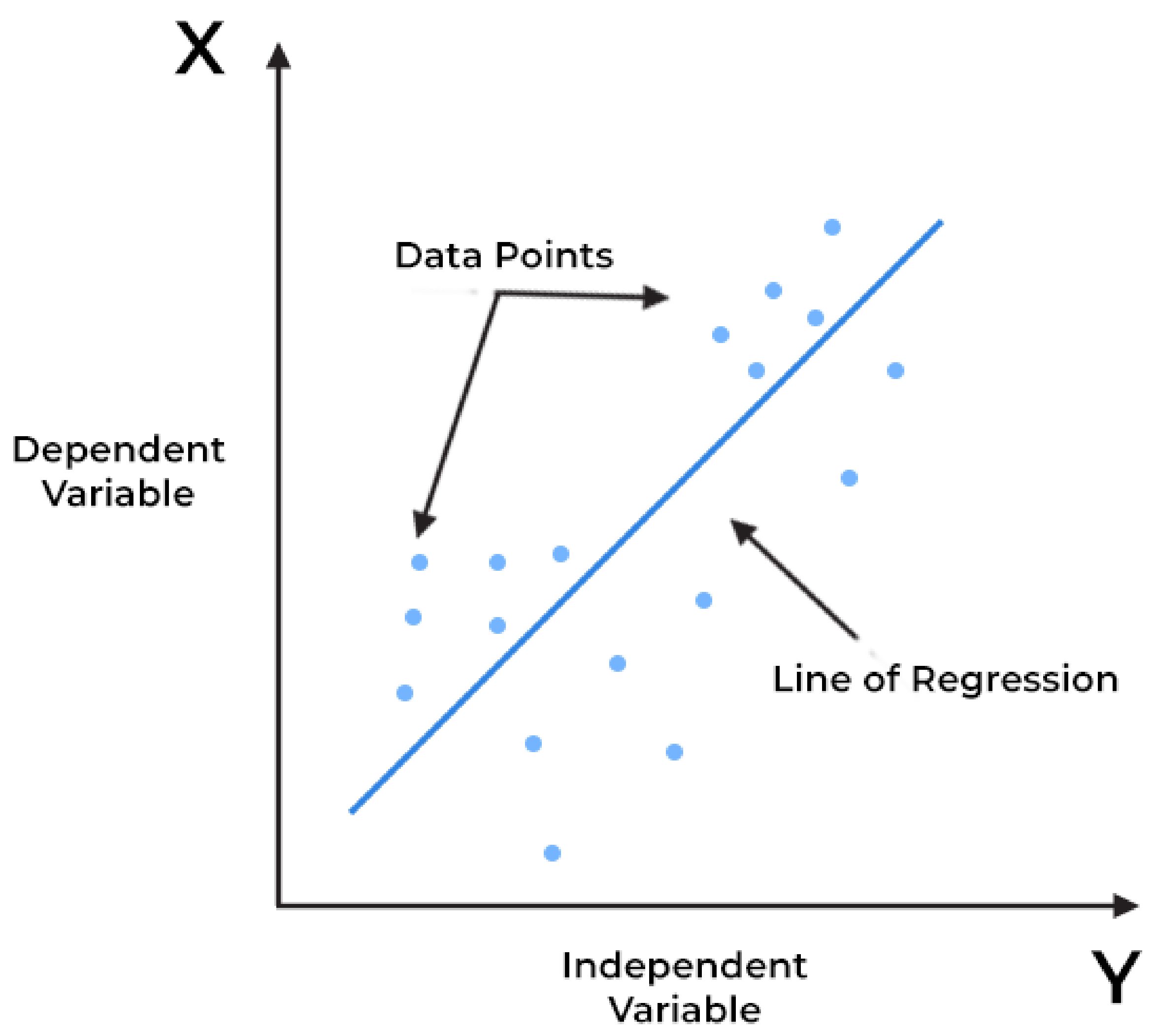
Principal Component Analysis

Alexandr Gavrilko
MLA Course for CS-2227

Short review of Practice 3: Linear regression

Linear Regression algorithm is a regression model that establishes a linear relationship between the independent variable(s) and the dependent variable. The algorithm aims to find the best-fit line that minimizes the sum of squared differences between the observed and predicted values.

Eager learning algorithm - there is training phase, during which algorithm find best values for parameters (coefficients)



Parametric algorithm - algorithm assume linear relationships between independent attributes and dependent attribute

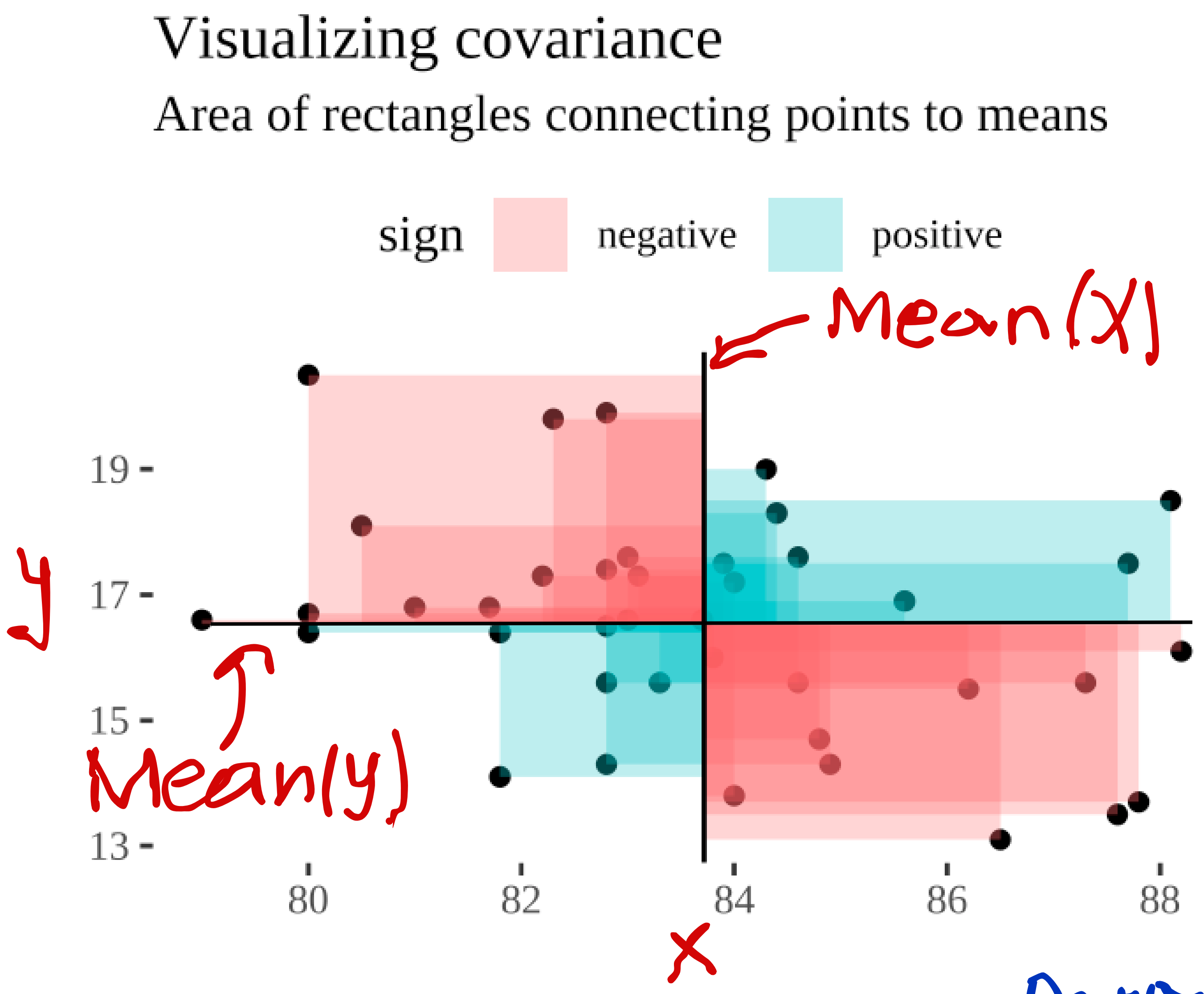
Formula: $$y = \underbrace{a}_{\text{Intercept}} + \underbrace{bX}_{\text{slope}}$$

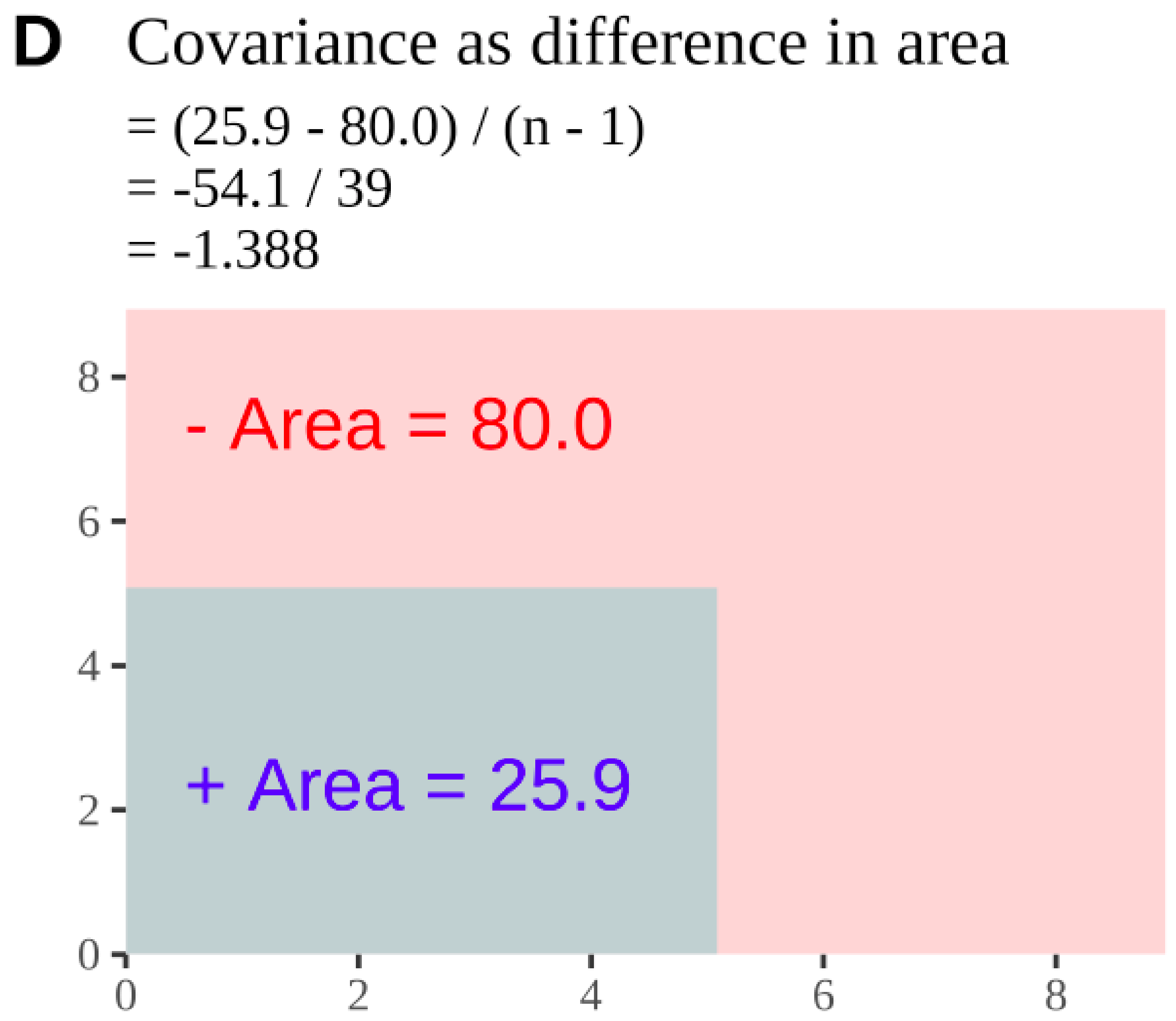
Types of regressions:

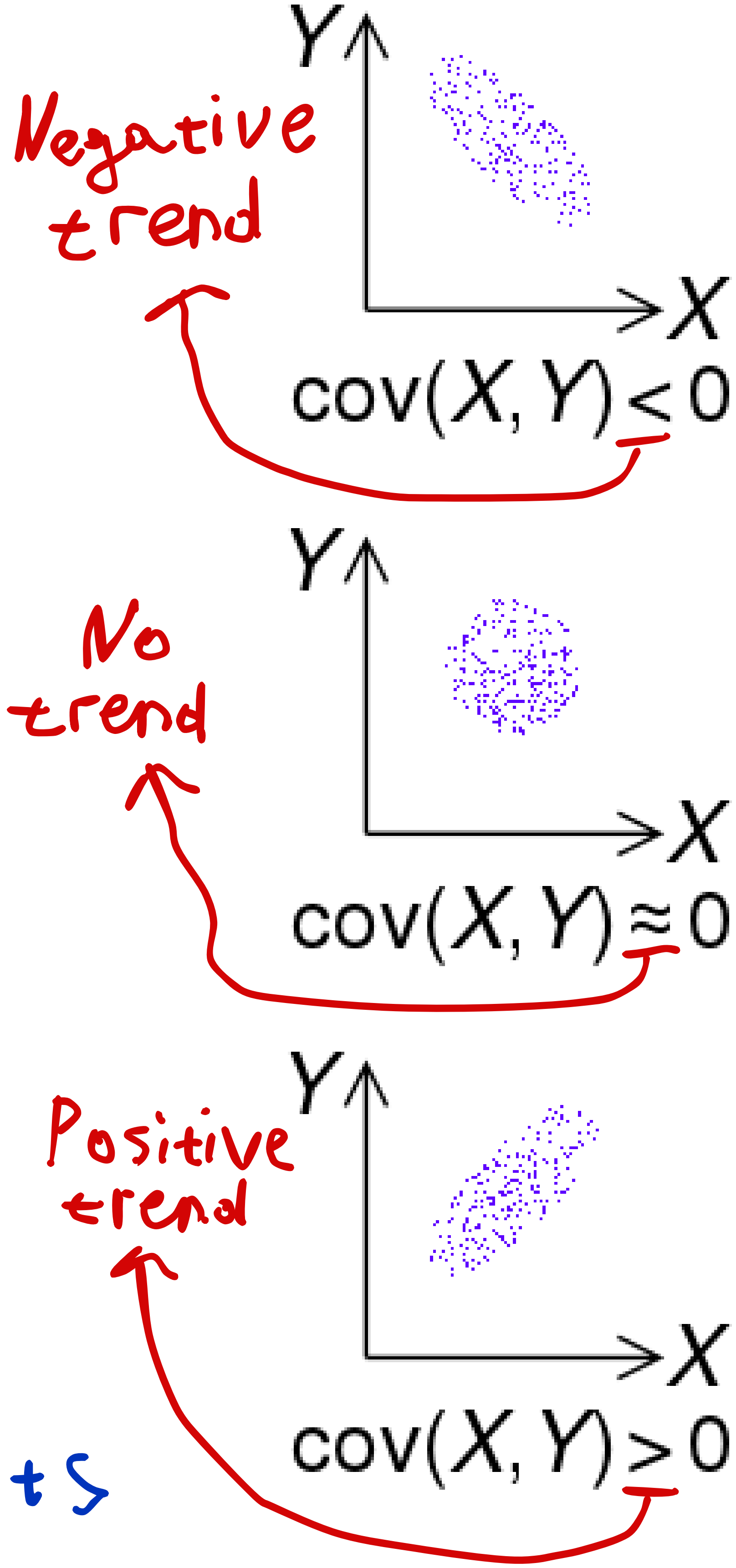
- Linear regression
- Multivariable regression - (multiple independent attributes)
- Polynomial regression - (more coefficients, regression line can be curved)
- Logistic regression - (regression for classification)

Short review of Practice 3: Covariance

Covariance is a measure of the relationship between two random variables. The metric evaluates how much - to what extent - the variables change together.







Data Point

Mean

$$Cov(x,y) = \frac{\Sigma\,(x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

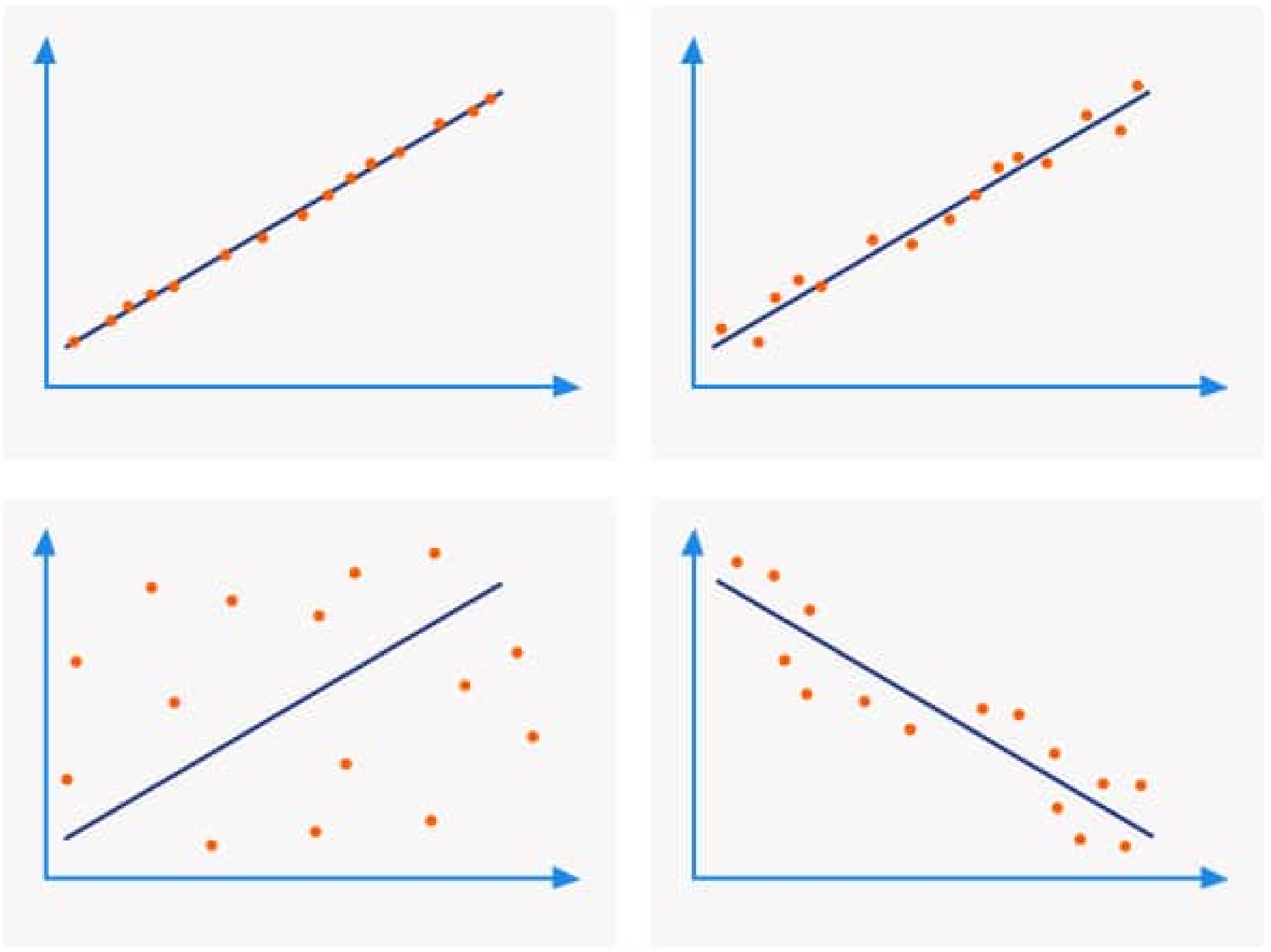
N ← Num. of data points

Short review of Practice 3: Correlation

Correlation is a statistical measure that expresses the extent to which two variables are linearly related (meaning they change together at a constant rate).

Limitation of Covariance: Covariance values can range from negative infinity to positive infinity, making interpretation difficult.

$Cov(X,Y) = \boxed{1000}$ Big or small?



1. Large positive correlation

$Corr \approx 1$

2. Medium positive correlation

$Corr \approx 0.7$

3. Small negative correlation

$Corr \approx -0.3$

4. Weak / no correlation

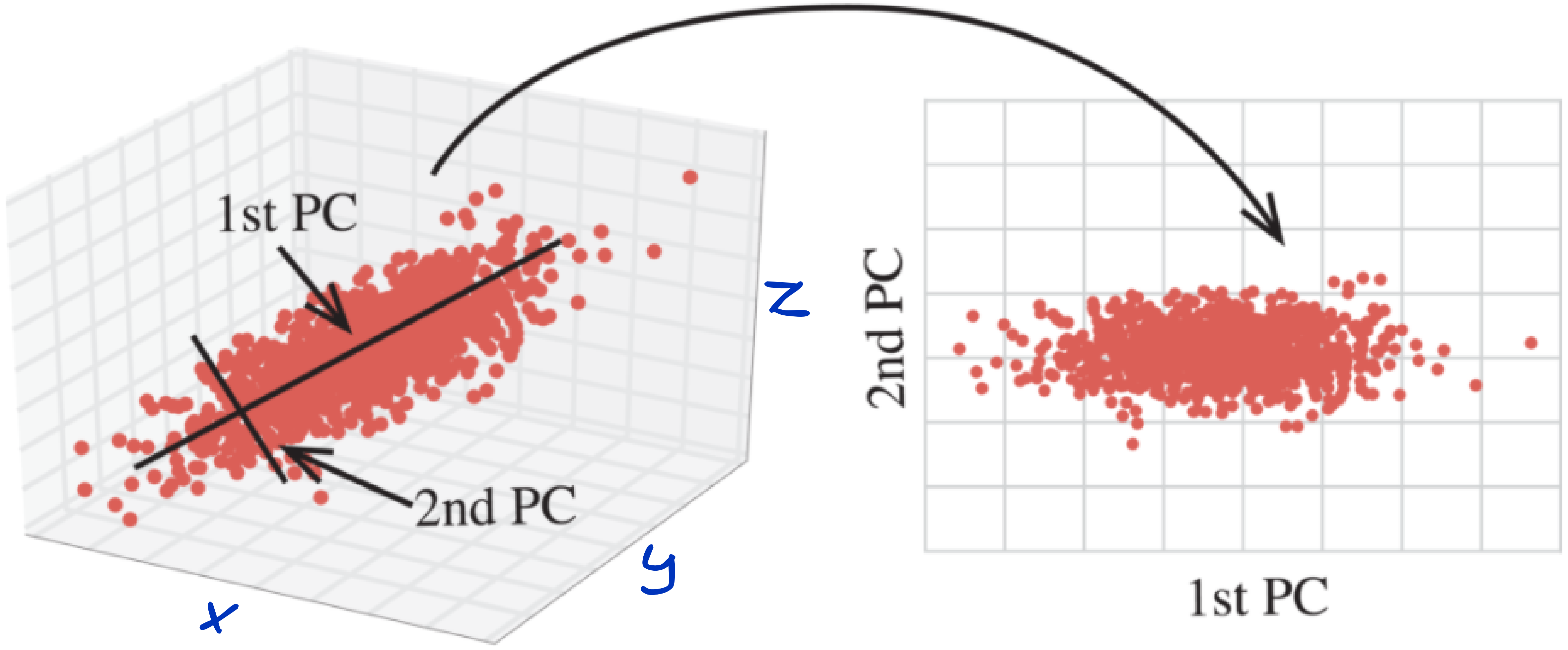
$Corr \approx 0$

$$Correlation = \frac{Cov\,(x, y)}{\sigma x * \sigma y}$$

By dividing by the product of the standard deviations of X and Y, the correlation value is limited to between -1 and 1.

Principal Component Analysis (PCA): Motivation

PCA is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.



Example: 3D to 2D

Principal Component Analysis (PCA): Motivation

Reason 1: Extract only most important information

PCA helps mitigate the curse of dimensionality by reducing the number of features while retaining the most important information.

Reason 2: Improved Model Performance

With fewer features, models are often computationally more efficient and require less memory. Training and testing times for machine learning models can be significantly reduced.

Reason 3: Data Visualization

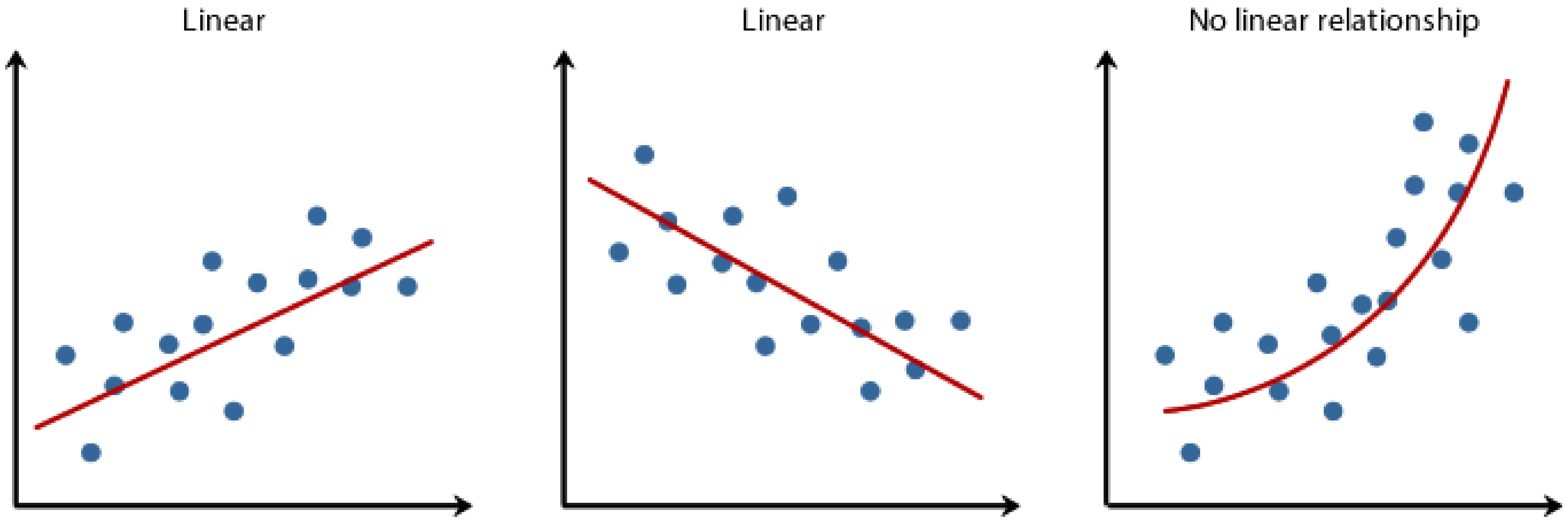
Reducing the dimensionality of the data to two or three principal components allows for easier visualization of complex datasets. This is particularly useful when exploring and understanding the structure of the data.

Reason 4: Data Compression

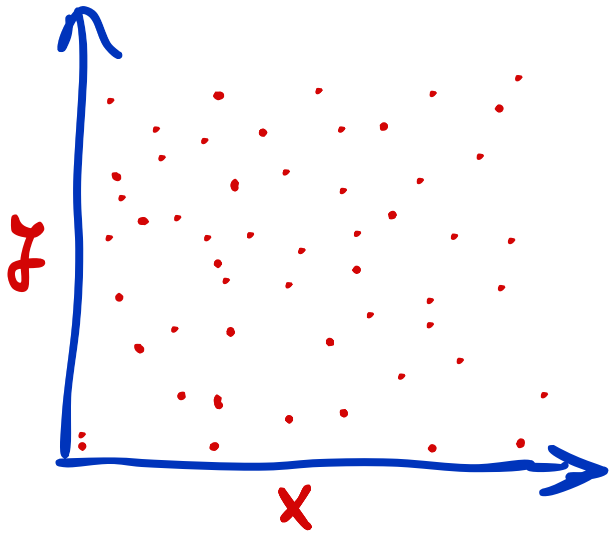
PCA allows for the compression of data, making it more storage-efficient. This is particularly valuable when working with large datasets.

Principal Component Analysis (PCA): Motivation

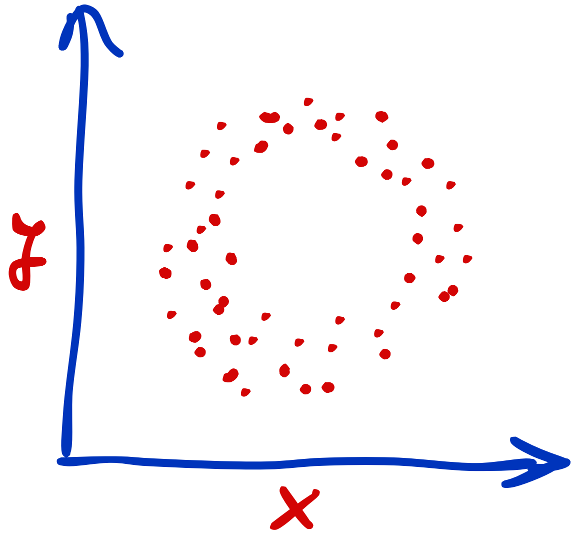
When not to use PCA? - When relation between data **is not linear**



Extra examples:



Correlation = 0



Non-linear relation-ships

Principal Component Analysis (PCA): Algorithm

Step 1. Preprocess data

Ensure that the data is properly pre-processed and cleaned, as PCA is sensitive to the scaling of the data.

Standardization: Standard Scaler

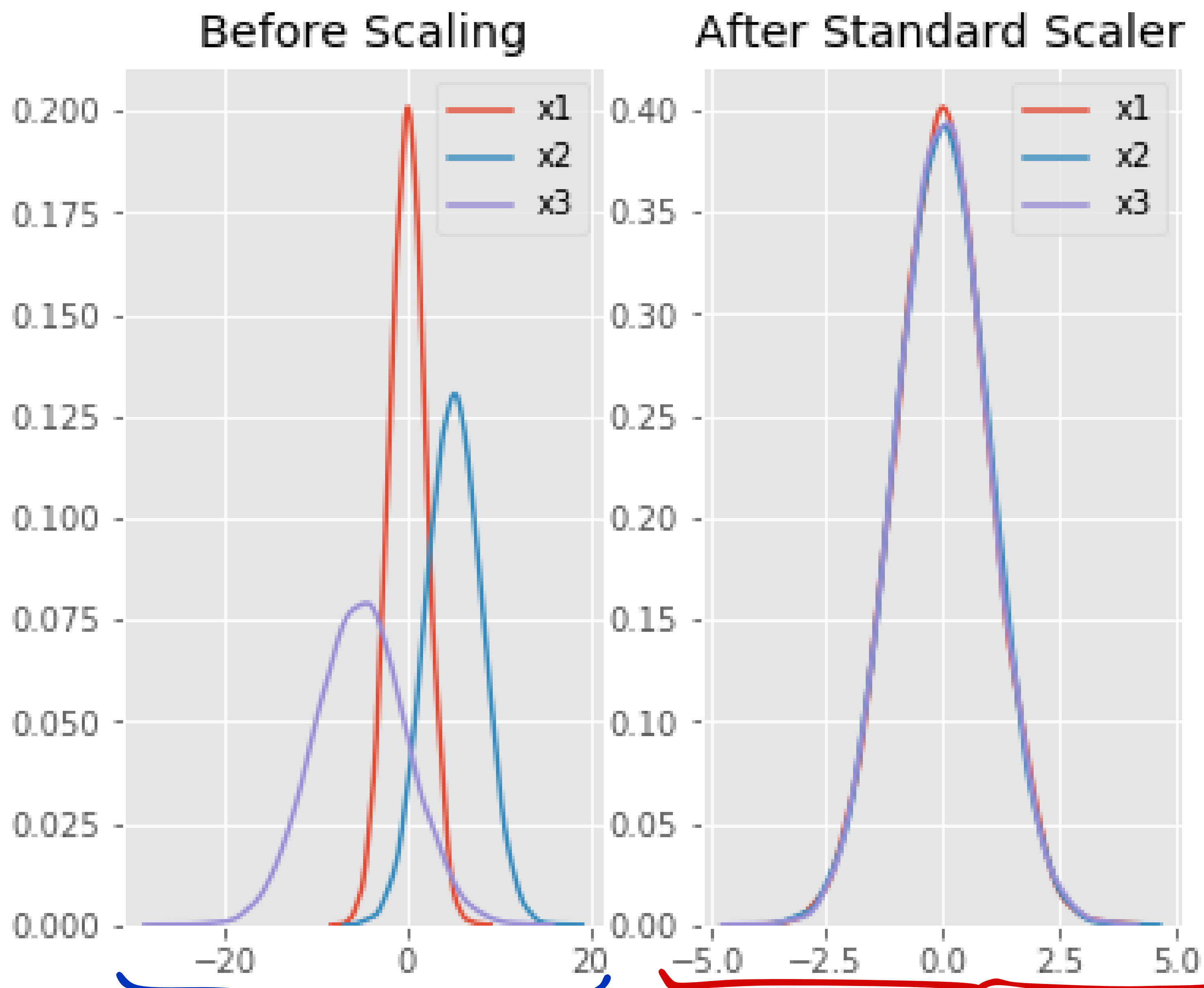
$$z = \frac{x-\mu}{\sigma}$$

with mean:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$

and standard deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}$$



Different μ
Different σ

Identical μ
Identical σ

Principal Component Analysis (PCA): Algorithm

Step 2. Compute Covariance matrix

The covariance matrix is a symmetric matrix that describes the relationship between variables in the data set.

Data Point

Mean

$$\mathrm{Cov}(x,y) = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

← Num. of data points

Covariance matrix for 2 attributes

$$\begin{array}{c c} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} x \\ y \end{array} & \begin{bmatrix} var(x) & cov(x,y) \\ cov(x,y) & var(y) \end{bmatrix} \end{array}$$

Covariance matrix for 3 attributes

$$\begin{array}{c c} & \begin{array}{ccc} x & y & z \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & \begin{bmatrix} var(x) & cov(x,y) & cov(x,z) \\ cov(x,y) & var(y) & cov(y,z) \\ cov(x,z) & cov(y,z) & var(z) \end{bmatrix} \end{array}$$

Common form:

$$\begin{bmatrix} Var(x_1) & \cdots & Cov(x_n, x_1) \\ \vdots & \ddots & \vdots \\ Cov(x_1, x_n) & \cdots & Var(x_n) \end{bmatrix}$$

Principal Component Analysis (PCA): Algorithm

Step 3. Compute the eigenvectors and eigenvalues of the covariance matrix

The eigenvectors and eigenvalues of the covariance matrix give us information about the structure of the data.

$$\underbrace{\begin{bmatrix} Var(x) & Cov(x,y) \\ Cov(y,x) & Var(y) \end{bmatrix}}_{\text{Covariance matrix}} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}' = \underbrace{\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}}_{\text{Eigenvalues}} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Eigenvectors

<u>You can check the following article to refresh memory on how to find Eigenvectors and Eigenvalues</u>

Principal Component Analysis (PCA): Algorithm

Step 4. Select the top k eigenvectors

The top k eigenvectors are **the principal components** and are used to reduce the dimensionality of the data.

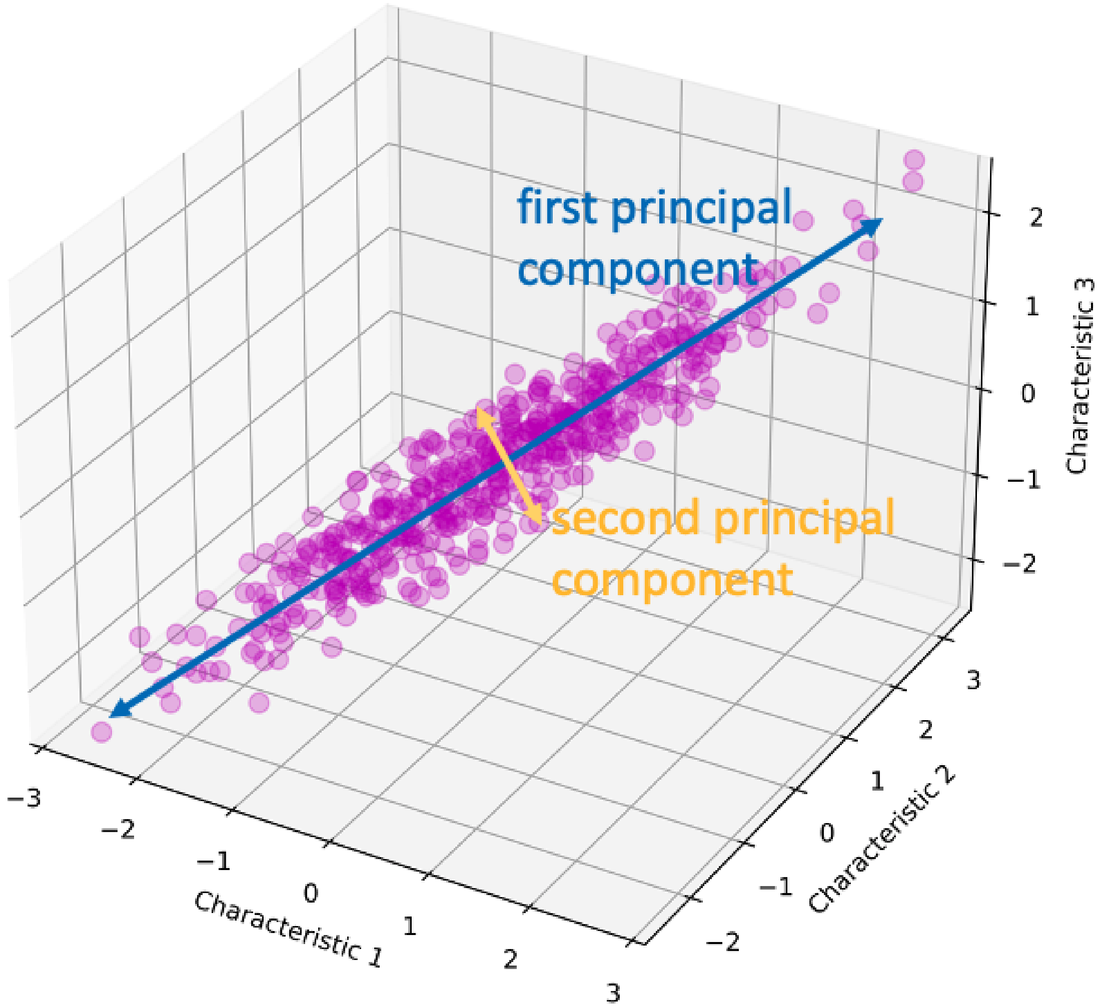
Desc

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Sort eigenvectors with respect to eigenvalues

Select top K vectors

Each eigenvector (principal component) reflects the variance of data. As they are higher in top, as they better reflect this variance. By extracting only top K vectors, we extract only most important information.



Principal Component Analysis (PCA): Algorithm

Step 5. Project the original data onto the principal components

The original data is transformed into a lower-dimensional space defined by the top k eigenvectors.

$$X_e = X^T e = \sum_{j=1}^{n} x_q e_j$$

Projected data

Original data

Eigenvectors

$$(5 \times 100)^T \cdot (5 \cdot 2) = (100 \times 2)$$