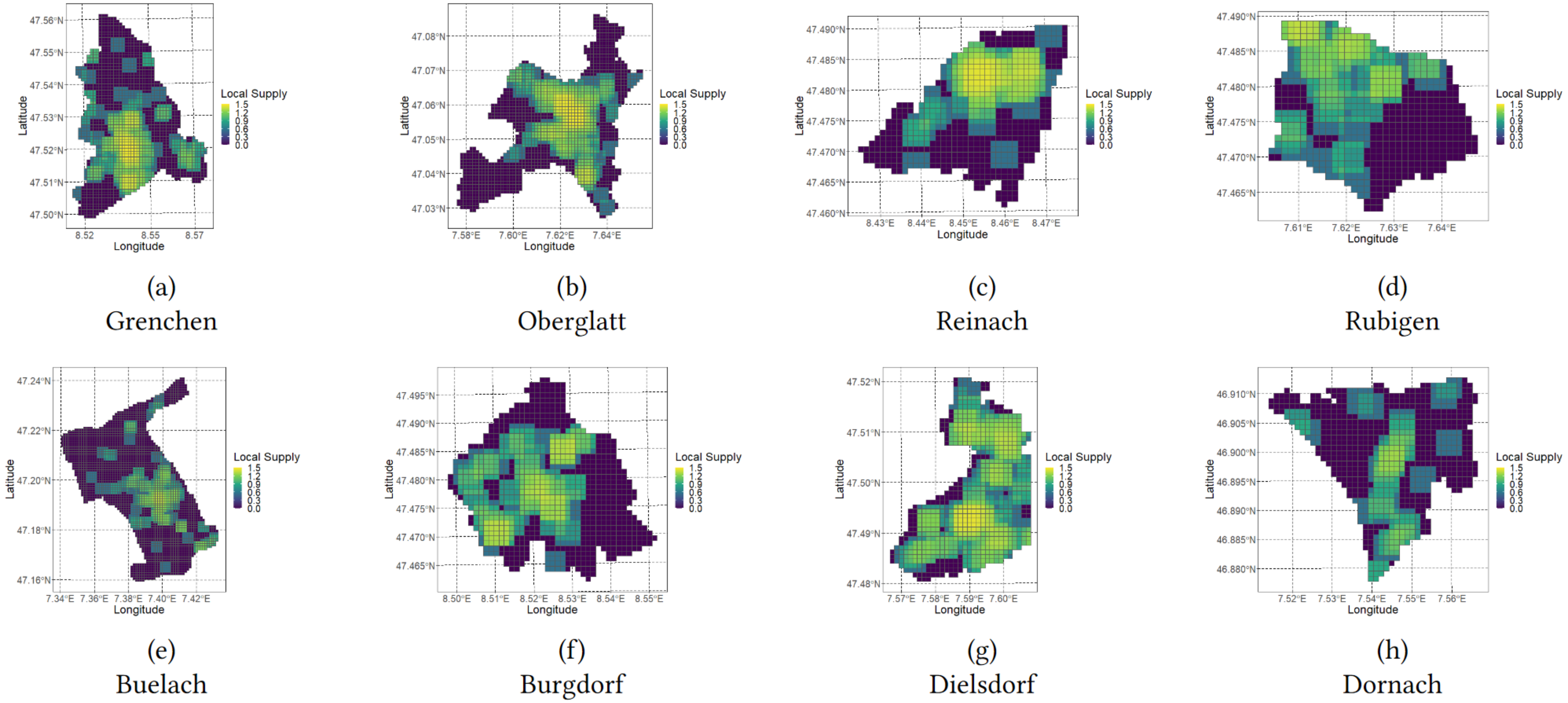
HSLU Hochschule Luzern

# CAS Machine Learning

# Unsupervised Learning: Clustering, Anomaly Detection, Dimensionality Reduction and Visualization



(a) Grenchen

(b) Oberglatt

(c) Reinach

(d) Rubigen

(e) Buelach

(f) Burgdorf

(g) Dielsdorf

(h) Dornach

**Dr. Yves Staudt**
PhD in Actuarial Sciences
yves.staudt@hslu.ch

# Agenda

1. Einführung
2. Kennenlernen Spiel
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Zusammenfassung

# Agenda

1. **Einführung**
2. Kennenlernen Spiel
3. Clustering
4. Data Preprocessing
5. K-Means Algorithmus
6. Determining Optimal Number of Clusters
7. Hierarchical Clustering Algorithmus
8. Kommunikation der Resultate
9. Dimensions Reduktionsverfahren
10. Principal Components Analysis
11. T-Distributed Stochastic Neighbor Embedding
12. Zusammenfassung

# Lernziel

Nach dem Kurs sind die Teilnehmende in der Lage:

- Merkmale nach Gleichheiten und Unterschiede zu bestimmen.
- Unterschiedsmasse zur Messung von Ähnlichkeiten zu beschreiben.
- Cluster Verfahren zu beschreiben, zu unterscheiden und anzuwenden.
- Cluster Resultate zu interpretieren.
- Dimension Reduktionsverfahren zu beschreiben, zu unterscheiden und anzuwenden.

# Dozent

Profil: Dr. Yves Staudt

Ursprung: Luxemburg

Erfahrung: Umfangreiche praktische Erfahrung in Datenanalysen, Data-Mining, Machine and Deep Learning

Interesse: Bergen, Fotografie und Kochen

Sozial: Ein sozialer Austausch ist mir wichtig.

# Unterrichtsplanung

| Uhrzeit | Thema |
|---|---|
| 9:15 – 10:00 | Einführung und Unterschiede bestimmen |
| 10:05 – 10:50 | Clustering, K-Means and Determining Optimal Number of Classes |
| 11:00 – 11:45 | Application Clustering |
| 11:40 -12:35 | Hierarchical Clustering Algorithmus |
| 12:35-13:40 | Mittagspause |
| 13:40 – 14:25 | Kommunikation und Interpretation der Resultate |
| 14:30 – 15:15 | Dimension Reduktionsverfahren, Principal Component Analys |
| 15:30 – 16:15 | t-Distributed Stochastic Neighbor Embedding |
| 16:15 – 16:45 | Zusammenfassung und Abschluss |

# Code Notebooks

Code Notebooks von Python werden über

- Google Drive oder
- Ilias zur Verfügung gestellt

Bei der Nutzung von Google Drive bitte eine Kopie erstellen.

Lösungen werden Schrittweise auf Ilias hochgeladen

# Agenda

1. Einführung
2. **Kennenlernen Spiel**
3. Clustering
4. Data Preprocessing
5. K-Means Algorithmus
6. Determining Optimal Number of Clusters
7. Hierarchical Clustering Algorithmus
8. Kommunikation der Resultate
9. Dimensions Reduktionsverfahren
10. Principal Components Analysis
11. T-Distributed Stochastic Neighbor Embedding
12. Zusammenfassung

# Spielregeln

- Interaktives Spiel
- Wir bewegen uns im Klassenraum
- Fragen zu:
    - ➢ Demografie
    - ➢ Arbeitsumfeld
    - ➢ Kenntnissen in Mathematik

Idee des Spiels: Nach jeder Frage bewegen wir uns so in der Klasse, dass wir uns den Teilnehmenden nähern, welche ähnliche oder gleiche Antworten auf die Frage habe.

Schwierigkeit: In der Fortsetzung der Fragen, versuchen wir die vorher beantworteten Fragen zu berücksichtigen.

Wenn keine Fragen sind dann geht es los.

# Frage 1

AI Appenzell Innerrhoden
AR Appenzell Ausserrhoden
BL Basel-Landschaft
BS Basel-Stadt
NW Nidwalden
OW Obwalden
SH Schaffhausen
SO Solothurn
DEUTSCHLAND
SH
BS
Thurgau
BL
Aargau
Zürich
Jura
SO
AR
AI
FRANKREICH
LIECHTEN-
STEIN
St. Gallen
Zug
ÖSTERREICH
Luzern
Schwyz
Neuen-
burg
Glarus
NW
Bern
OW
Uri
Freiburg
Waadt
Graubünden
Tessin
Genf
Wallis
ITALIEN

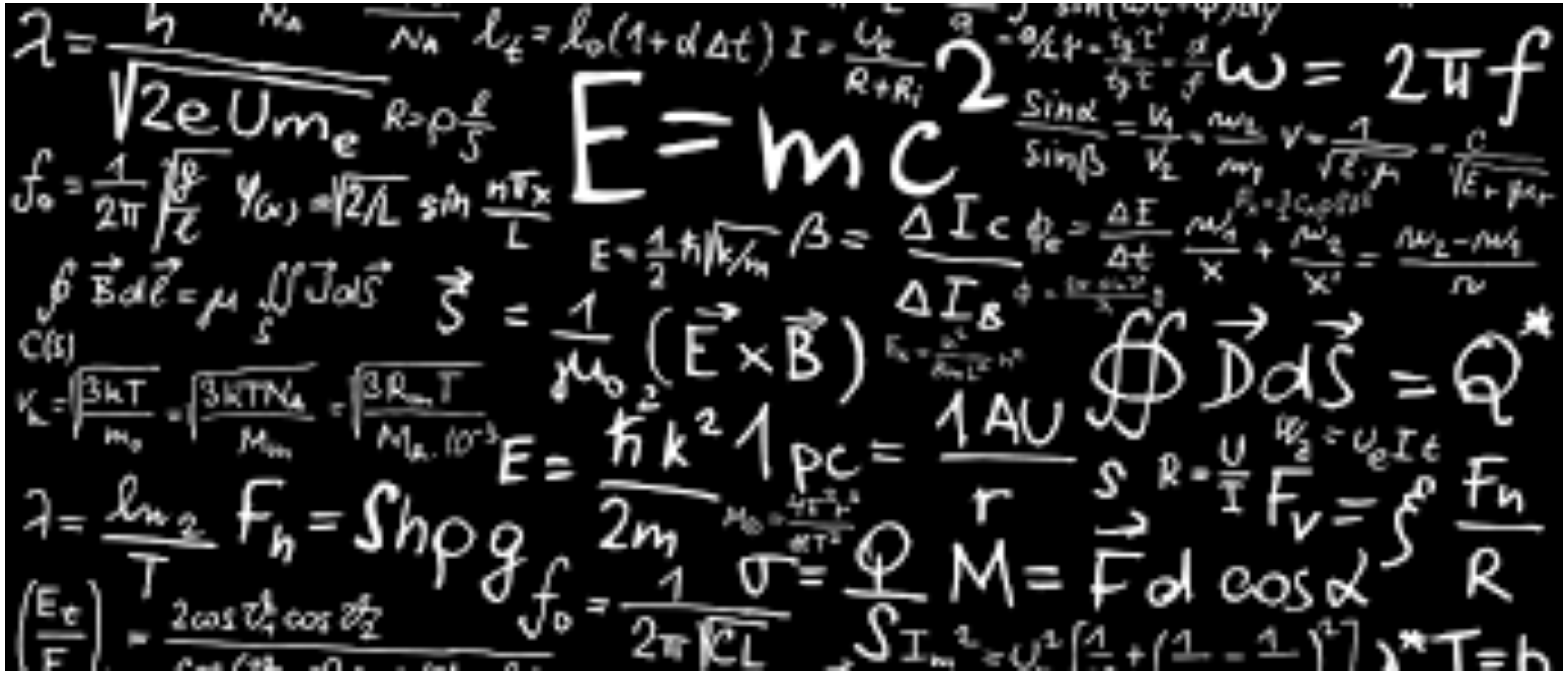
ISLAND
FINNLAND
NORWEGEN
Oslo
Helsinki
Sankt Petersburg
Stockholm
SCHWEDEN
ESTLAND
RUSSLAND
Moskau
LETTLAND
DÄNEMARK
Kopenhagen
Dublin
IRLAND
LITAUEN
VEREINGTES
KÖNIGREICH
Hamburg
NIEDERLANDE
Amsterdam
WEISSRUSSLAND
London
Berlin
Warschau
POLEN
Brüssel
BELGIEN
DEUTSCHLAND
Kiew
Frankfurt
Paris
LUXEMBURG
Prag
TSCHECHIEN
UKRAINE
SLOWAKEI
München
Wien
Zürich
FRANKREICH
SCHWEIZ
ÖSTERREICH
Budapest
MOLDAWIEN
Genf
UNGARN
Mailand
SLOWENIEN
KROATIEN
RUMÄNIEN
PORTUGAL
Bukarest
ANDORRA
BOSNIEN
SERBIEN
ITALIEN
MONTENEGRO
BULGARIEN
Madrid
KOSOVO
Lissabon
Barcelona
SPANIEN
Rom
MACEDONIEN
Istanbul
Neapel
ALBANIEN
Ankara
TÜRKEI
GRIECHENLAND
Athen
Algier
Casablanca
WWW.FREEWORLDMAPS.NET
MALTA
ZYPERN

Wo wohnen Sie?

## Frage 2

Wie alt sind Sie?

## Frage 3



Wie schätzen Sie ihre Mathematikkentnisse ein?

# Erkenntnisse

Wie verbinden Sie das Spiel mit dem Thema des Kurses?

Was sind eure Erwartungen an den Kurs?

Schreiben Sie in einer Gruppe von drei die Erkenntnisse auf (10 Min)

# Distances

Euclidean Distance: $\|x - y\|_2 = \sum_{j=1}^{n} \left(x_i - y_j\right)^2$ (Yin et al., 2021)

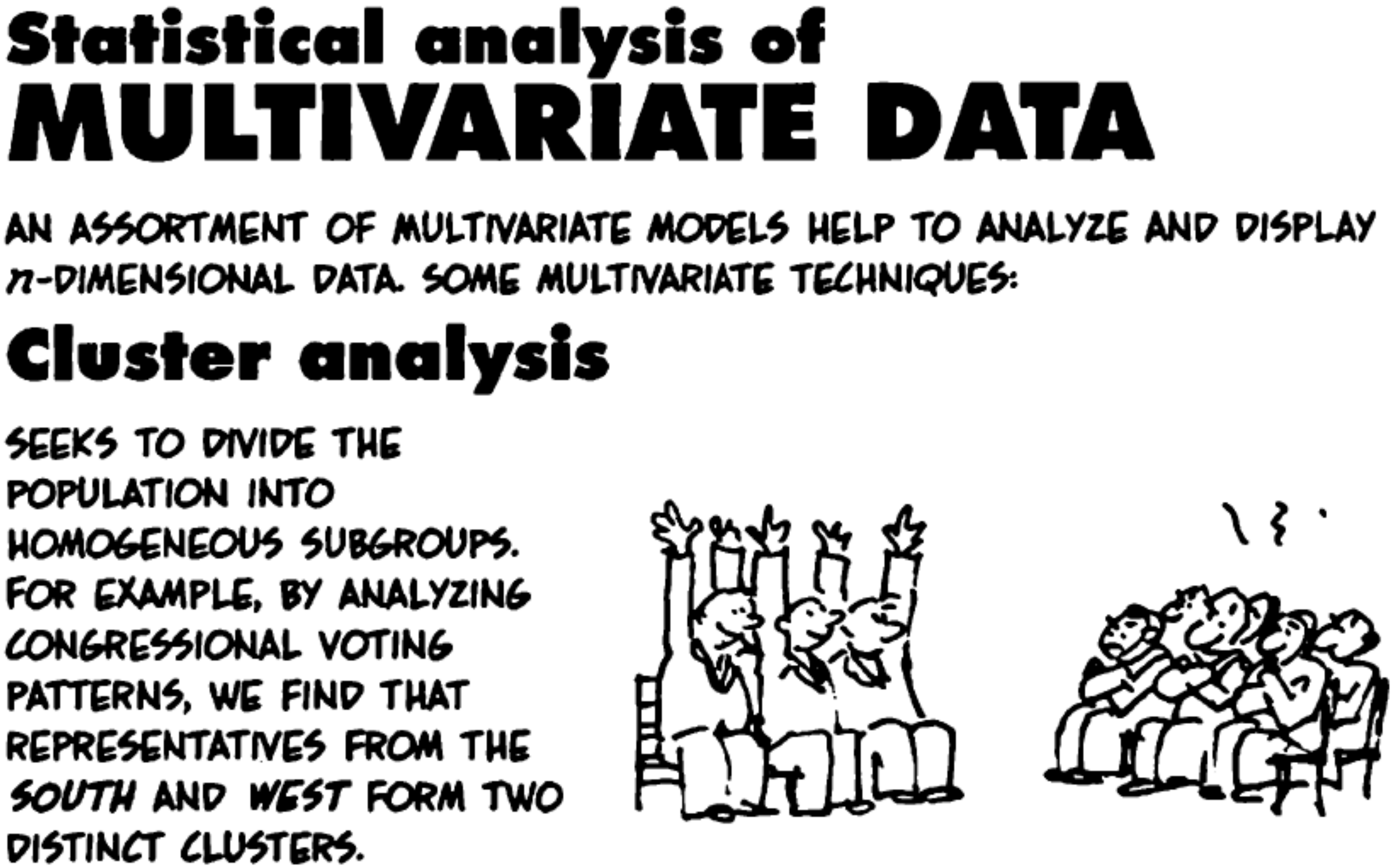
Simple Matching Distance: $D(x, y) = \begin{cases} 1, \ if\ x_j \neq y_j \\ 0, \ if\ x_j = y_j \end{cases}$ (Yin et al., 2021)

Dynamic Time Warping: $DTW(m, n) = \ |p_m - q_n| + \min \begin{pmatrix} DTW(m-1, n) \\ DTW(m-1, n-1) \\ DTW(m, n-1) \end{pmatrix}$ (Wong and Chung, 2019)

# Agenda

1. Einführung
2. Kennenlernen Spiel
3. **Clustering**
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Zusammenfassung

# Cluster Analysis Introduction



**Statistical analysis of**
**MULTIVARIATE DATA**

AN ASSORTMENT OF MULTIVARIATE MODELS HELP TO ANALYZE AND DISPLAY $n$-DIMENSIONAL DATA. SOME MULTIVARIATE TECHNIQUES:

**Cluster analysis**

SEEKS TO DIVIDE THE POPULATION INTO HOMOGENEOUS SUBGROUPS. FOR EXAMPLE, BY ANALYZING CONGRESSIONAL VOTING PATTERNS, WE FIND THAT REPRESENTATIVES FROM THE *SOUTH* AND *WEST* FORM TWO DISTINCT CLUSTERS.

Darstellung der Clusteranalyse und deren Anwendung (Gonick and Smith, 2005).

# Goal of Clustering

- The **goal** of **clustering** is to group similar objects or data points together based on their inherent characteristics

- Clustering is a fundamental task in **unsupervised learning**

- Clustering identifies natural groupings or patterns within the data

- Objects within the same cluster are more similar to each other than to those in other clusters

- Clustering algorithms help in understanding the **underlying structure** or organization of the data, revealing insights, and supporting decision-making processes.

# Application of Clustering

Clustering has various **applications** across different domains including:

- ➢ customer segmentation
- ➢ document classification
- ➢ anomaly detection
- ➢ image analysis
- ➢ data mining
- ➢ …

(Generated by Chat GPT 1.1.2023)

# Anomaly Detection

The goal of anomaly detection is to identify **unusual** or **anomalous** patterns or observations in a dataset

Unusual patterns deviate significantly from the norm or expected behavior

Anomalies can be indicative of errors, outliers, fraud, or any unexpected behavior that requires attention or investigation

Clustering aims to group **similar** data points together

Anomaly detection focuses on identifying the data points that are **dissimilar** or different from the majority

Clustering can be used as a preprocessing step for anomaly detection

By clustering the data, we can establish a notion of what is considered normal or expected within each cluster

Any data point that does not belong to any cluster or deviates significantly from its assigned cluster can be flagged as an anomaly

# Aufgabe 1: Händisches Clustering

- Bilden Sie Gruppen von 2 bis 3 Teilnehmenden
- Gruppieren Sie die Bilder in drei Gruppen
- Dokumentieren Sie in Stichworten wie sie die Gruppierung vorgenommen haben

# Aufgabe 2: Merkmale Bestimmen

- Begeben Sie sich in die vorher gegebenen Gruppen
- Dokumentieren Sie für die Bilder vier Merkmale/Variablen für die Bilder
- Dokumentieren Sie ob sich die Gruppierung verändern oder gleich bleibt

# Agenda

1. Einführung
2. Kennenlernen Spiel
3. Clustering
4. **K-Means Algorithmus**
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Zusammenfassung

# K-means Clustering

**Task of clusters:** Partitioning the dataset into groups, called clusters.

**Goal:** To split up the data in such way that points within single clusters are very similar and points in different clusters are different.

**Algorithm** alternates between **two steps**:

1. Assigning each data point to the closest cluster center.
2. Setting each cluster center as the mean of the data point that are assigned to it.

The algorithm is finished when the assignment of instances to clusters no longer changes.

In k-means clustering the number of clusters k needs to be fixed by the analyst.

(Jamies et al., 2017; Kuhn and Johnson, 2016)

# Properties of k-means clustering

Let $C_1, C_2, \ldots, C_K$ denote sets containing indices of the observations in each cluster

The sets satisfies the following two properties:

1. $C_1 \cup C_2 \cup \cdots \cup C_k = \{1, \ldots, n\}$
2. $C_k \cap C_l = \emptyset \; \forall \; k \; \neq \; l$

(James et al., 2013)

# Optimization

**Idea:** A good clustering is one for which the within-cluster variation is as small as possible.

Within-cluster variation for cluster $C_k$ measures the difference between the observations within a cluster.

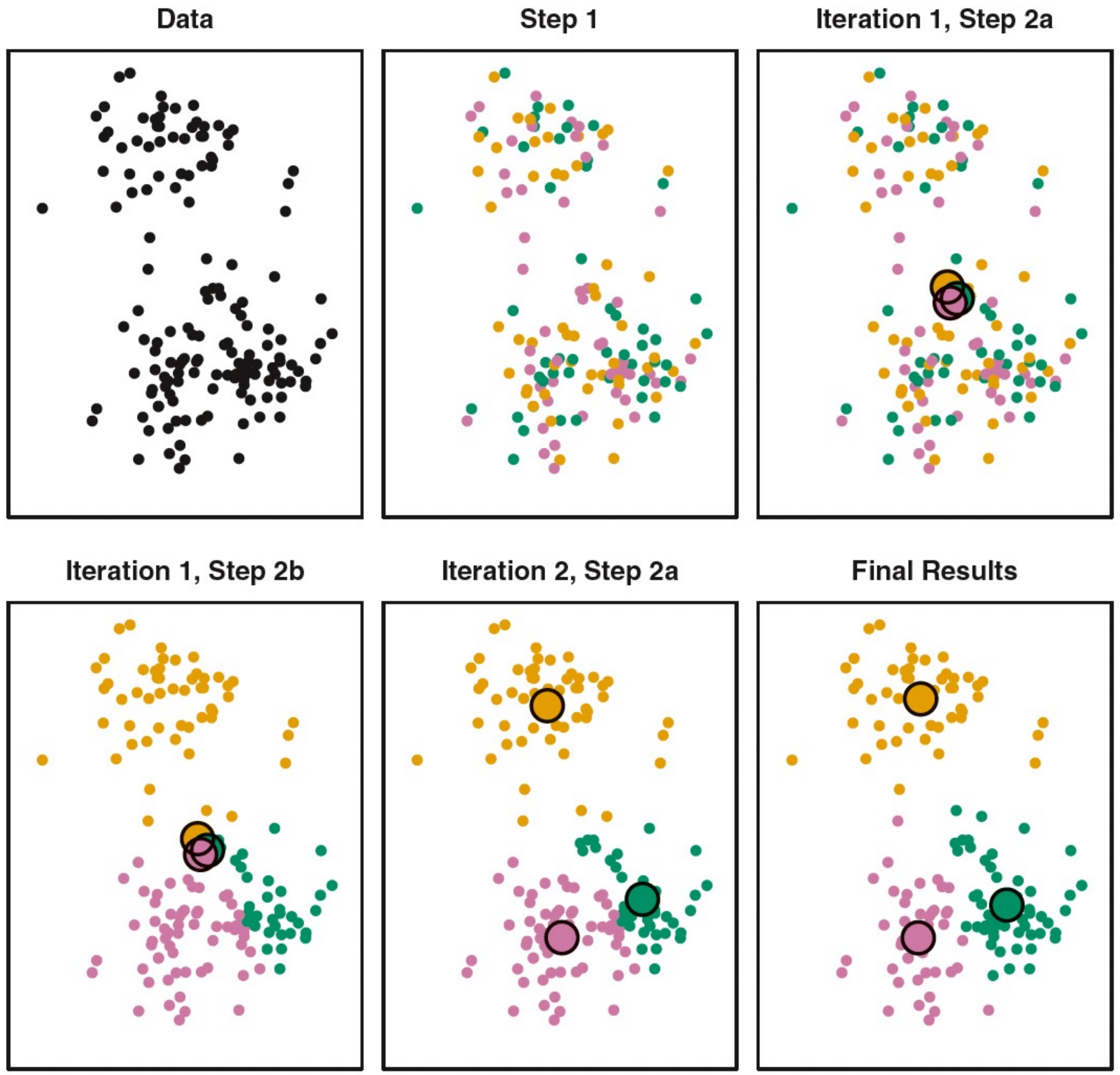
Goal: To minimize the within-cluster variation

$$minimize_{C_1,\ldots,C_k}\left\{\sum_{k=1}^{K} W(C_k)\right\}$$

$$minimize_{C_1,\ldots,C_k}\left\{\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}\left(x_{ij}-x_{i'j}\right)^2\right\}$$

(James et al., 2013)

# Representation of the k-means algorithm



Representation of the K-means algorithm (James et al. 2013)

# Algorithm k-means clustering

The algorithm is explained with the help of three clusters (**k = 3**)

1. Initialization: Declaring three (k=3) data points randomly as clusters center.
2. Assign Points: Each data point is assigned to the cluster center it is closest to. (Using similarity measure)
3. Cluster centers are updated to the mean of the assigned points (Recompute Centers)
4. Stop when no assignments are done anymore.

(Jamies et al., 2013; Kuhn and Johnson, 2013)

# Impact of scale

The scale of the variables has a huge impact in the algorithm.

To scale the data before application of k-means.

# Exercise

- Begeben Sie sich in Gruppen von 2 bis 3 Teilnehmenden
- Öffnen Sie das Code Notebook “clustering_k_means_exercise.ipynb”
- Führen Sie für den gegebenen Datensatz das k-Means Clustering für k=3 durch
- Dokumentieren Sie was Ihnen aufgefallen ist

# Agenda

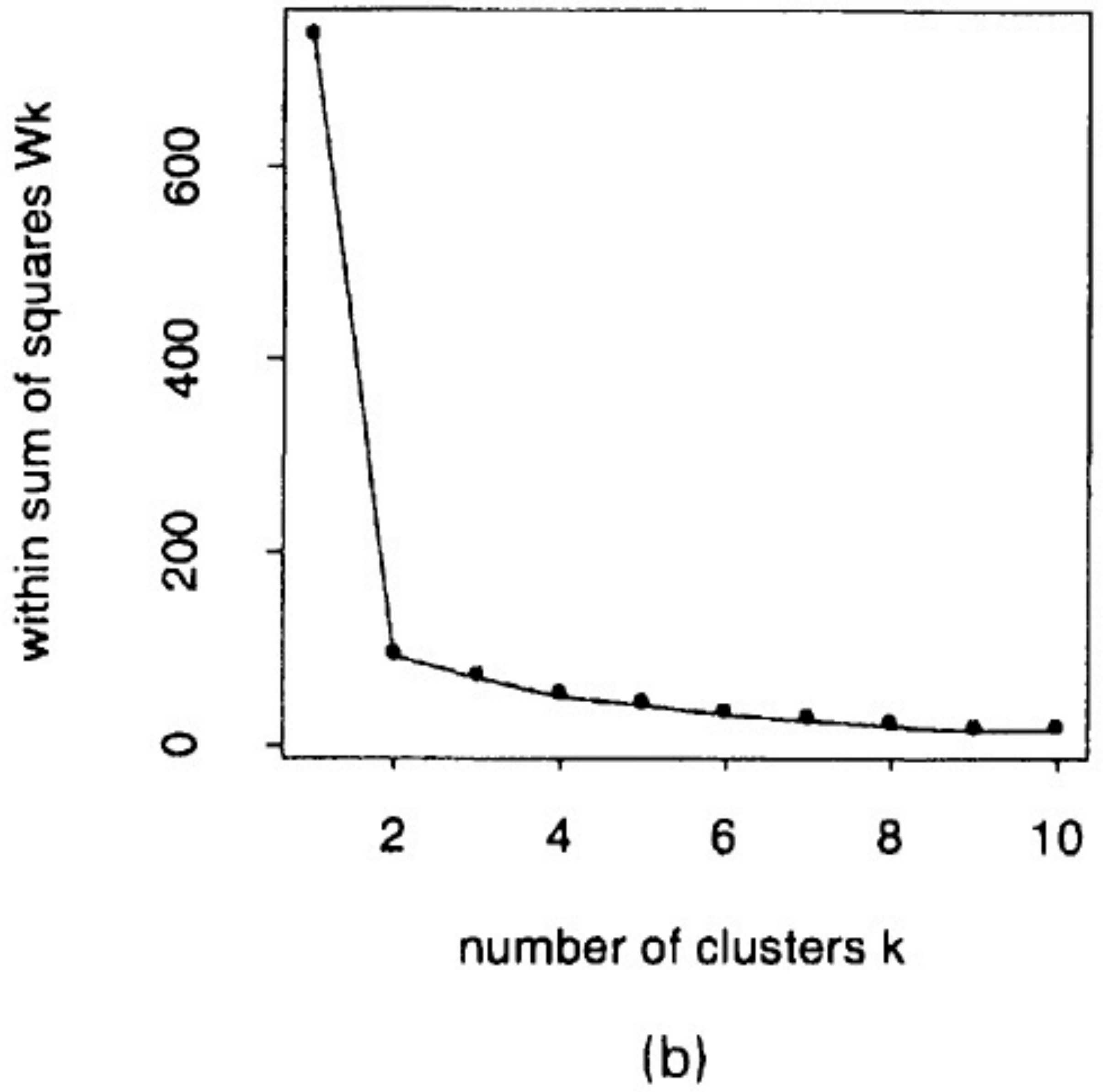
1. Einführung
2. Erwartungen an den Kurs
3. Clustering
4. K-Means Algorithmus
5. **Determining Optimal Number of Clusters**
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Zusammenfassung

# How to select optimal number of clusters

- But how determine the number of clusters?
- The seen mehtods from Machine Learning cannot be applied
- There exists several methods to assess optimal number of clusters
- We treat three mehtods
    1. Elbow method
    2. Silhouette
    3. GAP Statistic

# Elbow Method

- Most popular method
- Idea: Calculating the Within-Cluster-Sum-of Squared (WSS) for different number of clusters
- Elbow method decrease with increasing k



Representation of the elbow method (Tibshirani et al. 2001)

# Silhouette

- Silhouette coefficient tells us if inidividual points are correctly assigned to their clusters.
- Silhouette Coefficient for an obseration $i$ is defined as follows

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

–where $b(i)$ is the smallest average distance of point I to all points in any other cluster
–and $a(i)$ is the average distance of $i$ from all other points in its cluster

The Silhouette Coefficient for the data set is the average of the Silhouette Coefficient of individual points.

Meaning of $S(i)$ :

- $S(i)$ close to 0 means that the observation is between two clusters
- $S(i)$ close to -1 than the point should be assigne to the other cluster
- $S(i)$ close to 1, than the point belongs to the correct cluster

# GAP Statistic

Idea: To compare the cluster results with a null reference distribution of the data

The optimal number of cluster is the value of k for which $\log(W_k)$ falls the fathest below the curve of the reference distribution

$$Gap_n(k) = E_n^*\{\log(W_k)\} - \log(W_k)$$

- where $W_k$ is the within-cluster variation
- and $E_n^*\{\log(W_k)\}$ the expecation under a sample of size n form the reference distribution

The optimal $\hat{k}$ is obtained by maximizing $Gap_n(k)$

We assume a null model of a single component and reject it in favour of a k-component model

For each feature a unifromly distributed variable over the range of the observed values is created as reference distribution.

# Exercise

- Begeben Sie sich in Gruppen von 2 bis 3 Teilnehmenden

- Öffnen Sie das Code Notebook “clustering_k_means_optimizing_exercise.ipynb”

- Bestimmen Sie die optimale Anzahl Clusters

- Dokumentieren Sie ihre Resultate

# Feedback

Vielen Dank für euer Feedback auf folgender Seite

http://www.evaluationszielscheibe.ch/?disc=2a68db.

# Agenda

1. Einführung
2. Kennenlernen Spiel
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. **Hierarchical Clustering Algorithmus**
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Zusammenfassung

# Disadvantage k-means

- Number of clusters needs to be fixed

# Dendogram

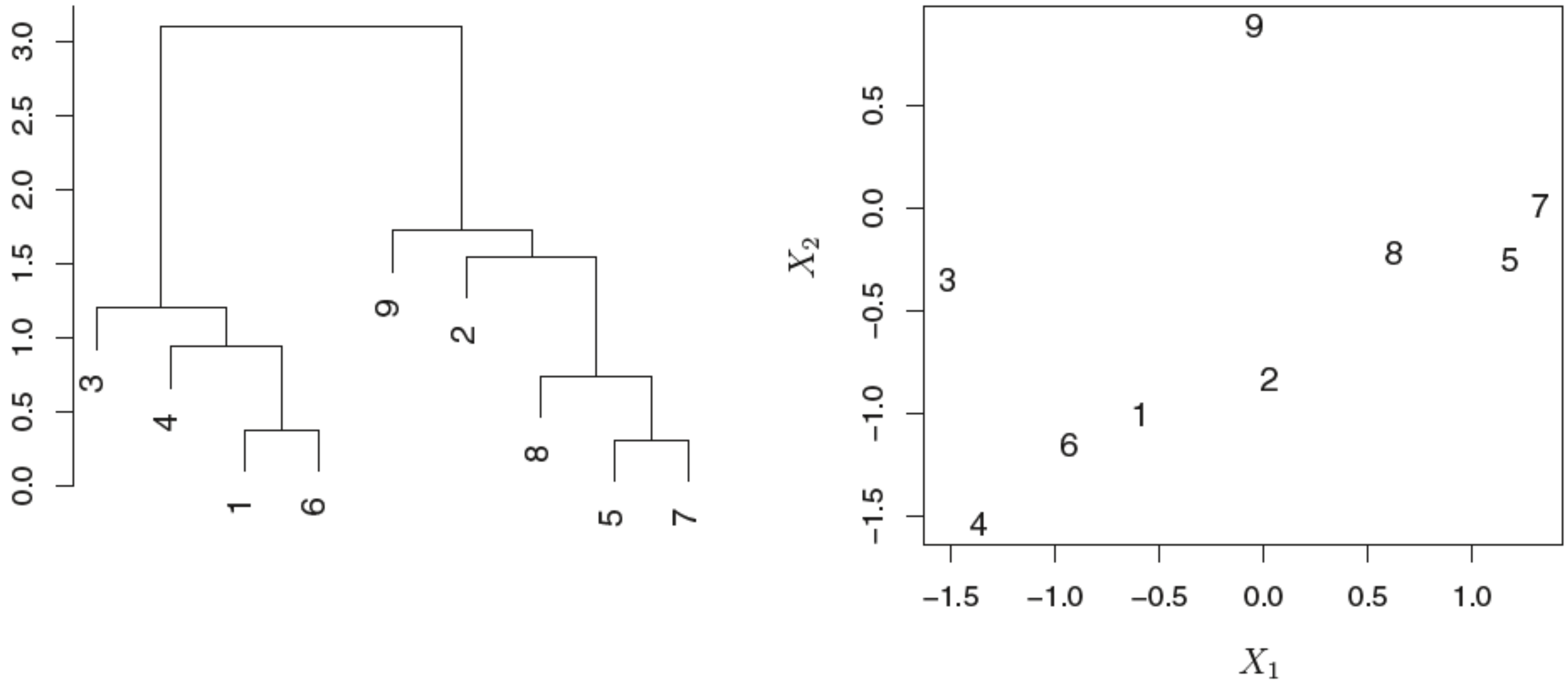
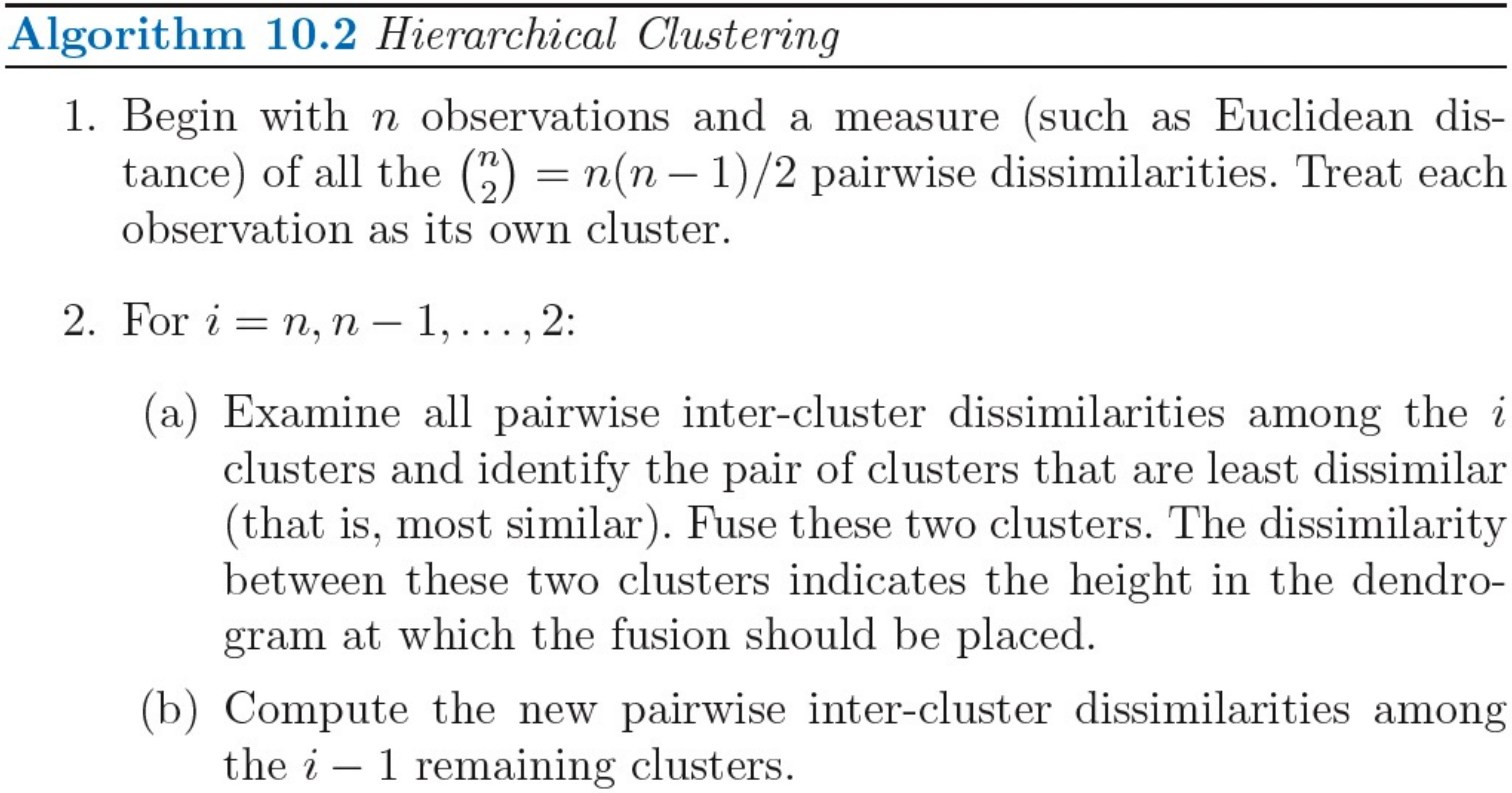


Illustration of a dendogram applied to a simple application (James et al., 2013).

# Algorithm Hierarchical Clustering



**Algorithm 10.2** *Hierarchical Clustering*

1. Begin with $n$ observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

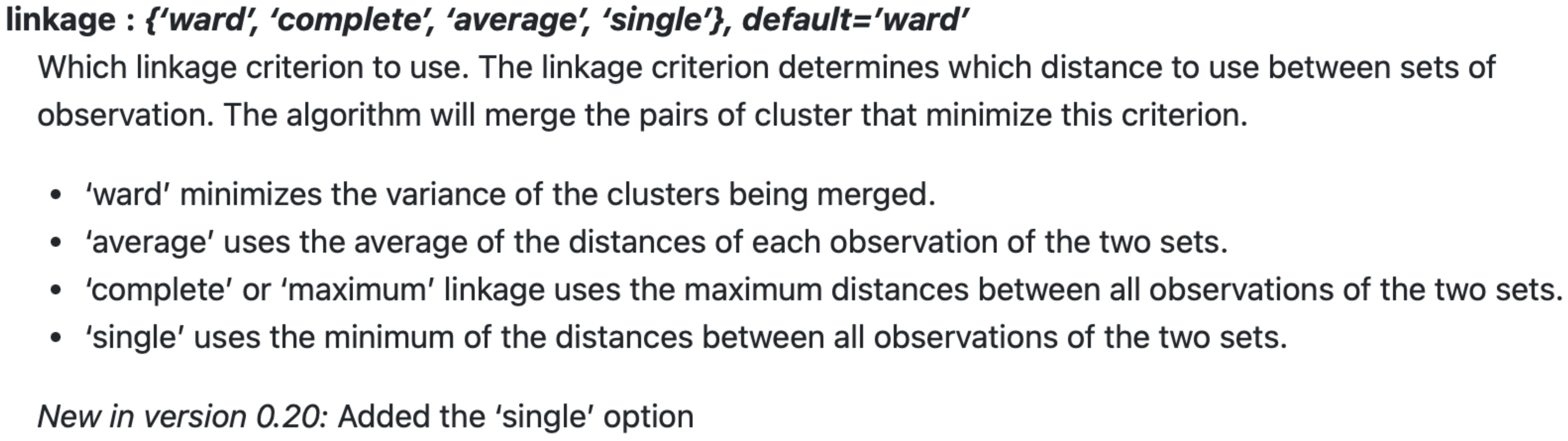
2. For $i = n, n-1, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the $i$ clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the $i - 1$ remaining clusters.

Illustration of the hierarchical clustering (James et al., 2013).

# Linkage functions in hierarchical clustering



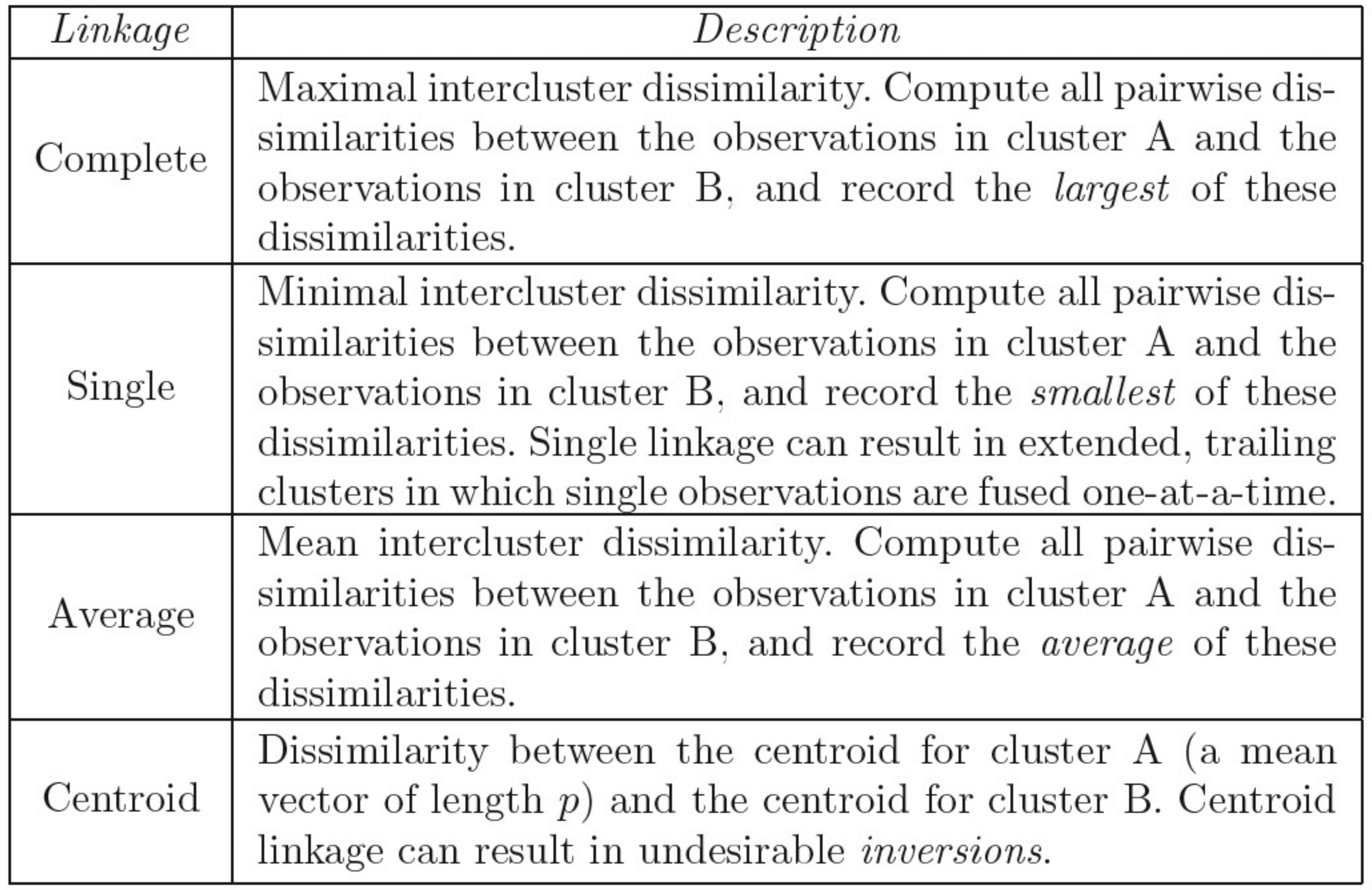
**linkage : *{'ward', 'complete', 'average', 'single'}, default='ward'***

Which linkage criterion to use. The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion.

- 'ward' minimizes the variance of the clusters being merged.
- 'average' uses the average of the distances of each observation of the two sets.
- 'complete' or 'maximum' linkage uses the maximum distances between all observations of the two sets.
- 'single' uses the minimum of the distances between all observations of the two sets.

*New in version 0.20:* Added the 'single' option

Illustration of the different linkage functions applied in hierarchical clustering in Scikit-Learn.



| *Linkage* | *Description* |
|---|---|
| Complete | Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities. |
| Single | Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time. |
| Average | Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities. |
| Centroid | Dissimilarity between the centroid for cluster A (a mean vector of length $p$) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*. |

Illustration of the different linkage functions applied in hierarchical clustering (James et al., 2013).

# Application example

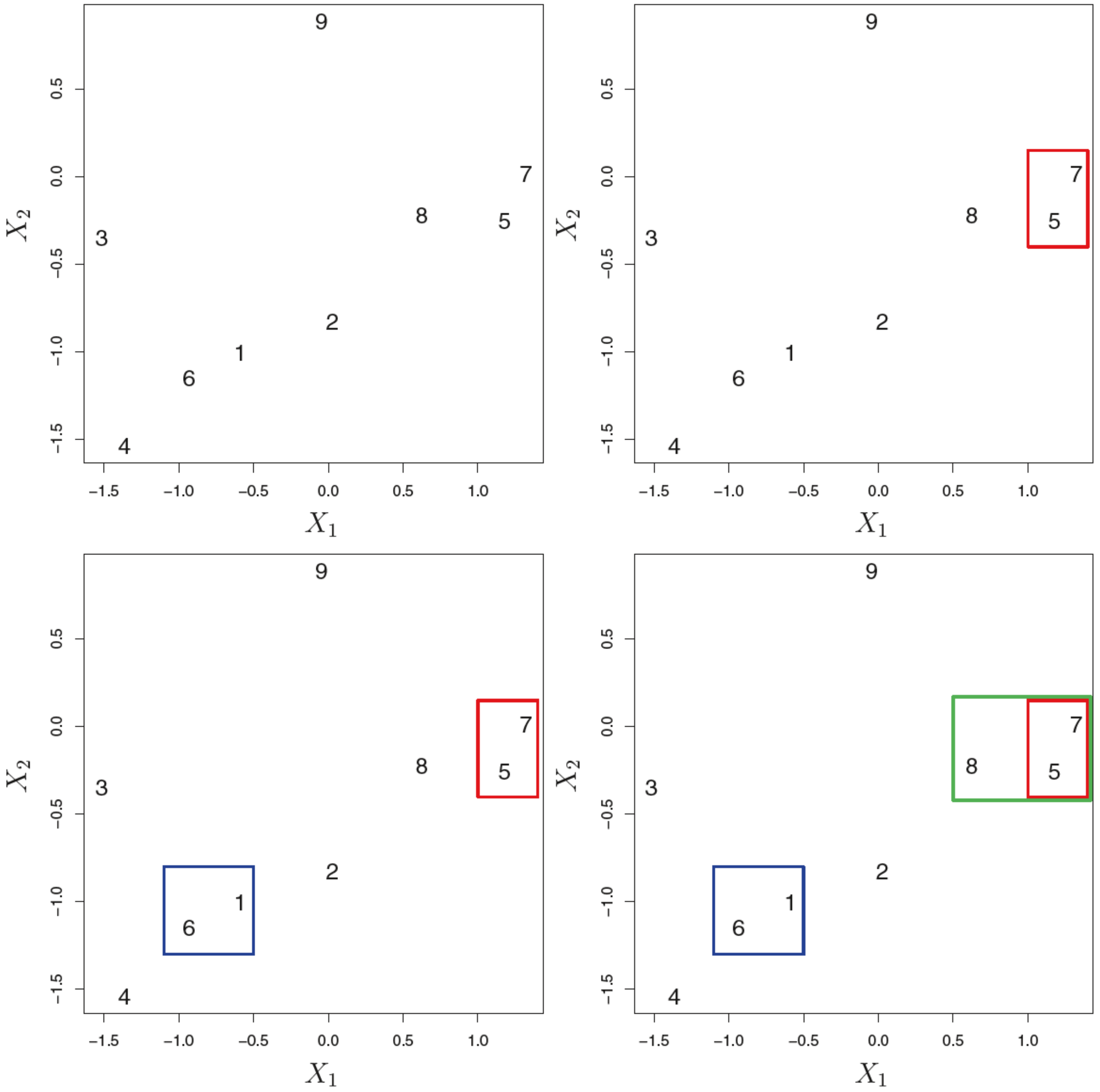


Illustration of an application of the hierarchical clustering algorithm (James et al., 2013).

# Difference due to the linkage function

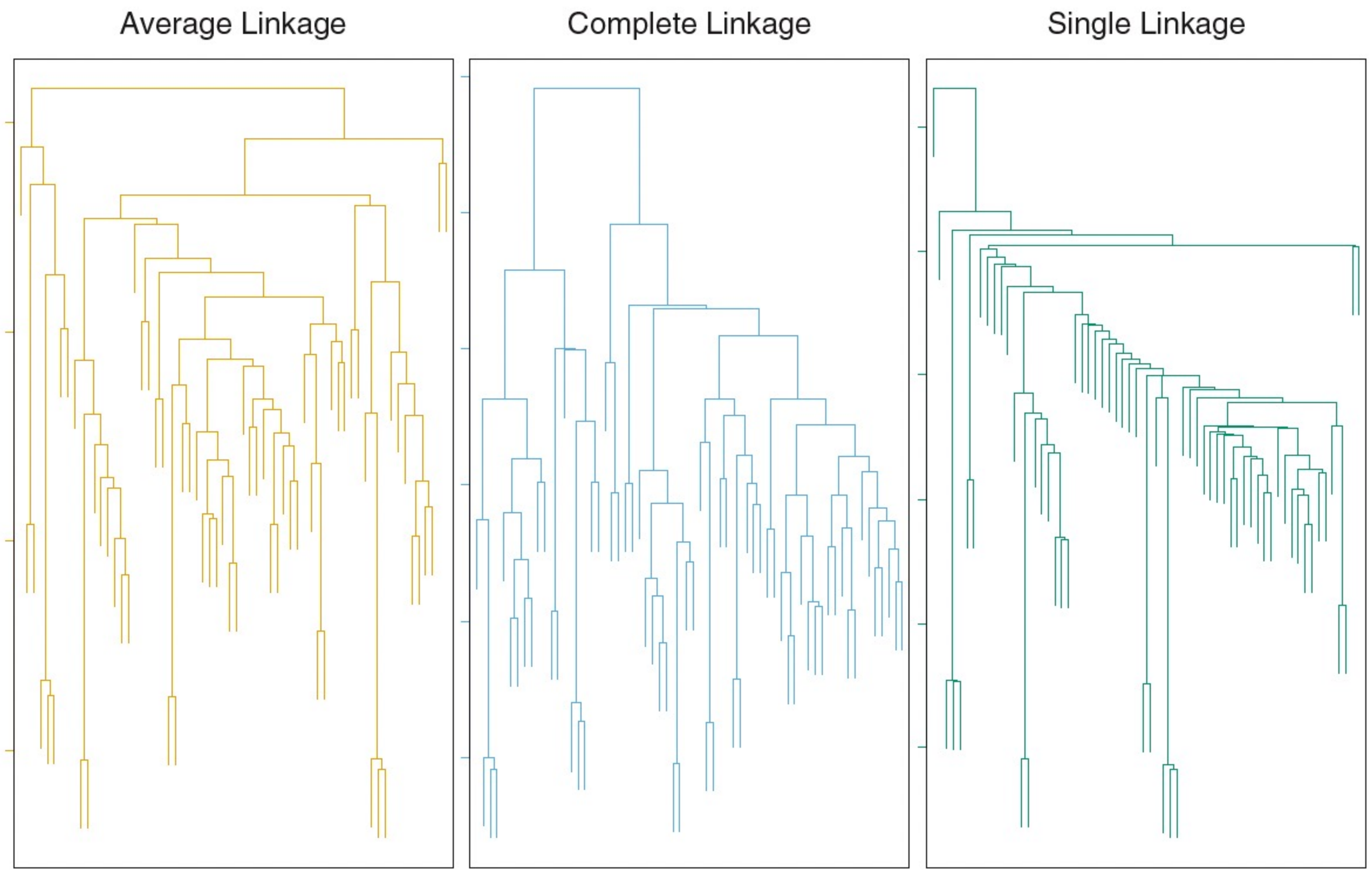


Illustration of differences in the dendogram due to the linkage function (James et al., 2013).

# Small Decisions with Big Consequences

- Standardization of data
- Use of dissamilartiy measure
- Use of linkage
- Number of clusters

# Exercise

- Begeben Sie sich in eine Gruppe von 2 bis 3 Teilnehmenden
- Öffnen Sie das Codenotebook “hiearchical_clustering_exercise.ipynb”
- Führen Sie das Hierarchical Clustering vorherigen optimale Clusteranzahl durch
- Dokumentieren Sie ihre Erkenntnisse

# Agenda

1. Einführung
2. Kennenlernen Spiel
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. **Kommunikation der Resultate**
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Zusammenfassung

# Wieso?

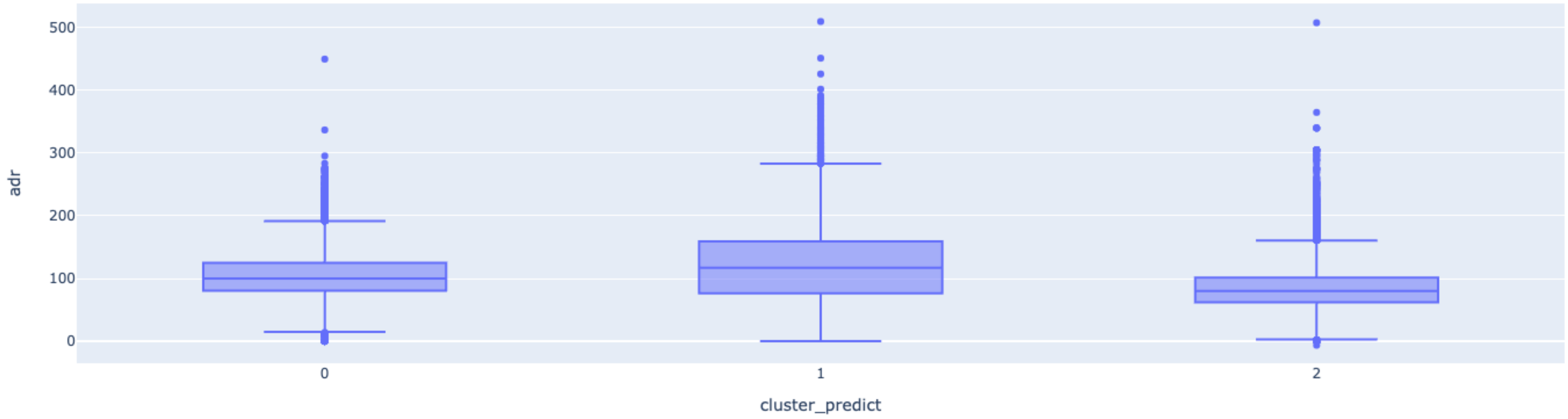
Wie unterscheiden sich die Gruppen?

Wie kann ich die Gruppen am besten unterscheiden?

Welche Schwierigkeit ist Ihnen aufgefallen?

# Möglicher Weg

- Die Resultate werden auf den skalierten Daten angezeigt
- Wir würden aber gerne den Bezug zu den Ursprungsdaten herleiten
- Lösung: Erhalten Cluster mit den Ursprungsdaten verbinden
- Cluster nach Merkmalen analysieren
- Beste Möglichkeit: Visualisierung und Deskriptive Statistiken



# Exercise

- Begeben Sie sich in Gruppen von 2 bis 3 Teilnehmenden
- Beschreiben Sie die Clusters
- Dokumentieren Sie wie sie die Gruppen beschreiben würden

# Agenda

1. Einführung
2. Erwartungen an den Kurs
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. **Dimensions Reduktionsverfahren**
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Zusammenfassung

# Challenge

- Many Machine Learning problems involve **thousands** or even **millions** of features for each trianing instance
- High number of dimensionality can have serveral **drawbacks**
  - ❖ High dimensionality makes training extremely **slow**
  - ❖ High dimensionality makes it harder to find a **good** solution
  - ❖ High dimensionality makes it harder to **interpret** the results
- The high dimensionality problems are known as **curse of dimensionality**
- Dimensionality reduction reduce the **number of features** considerably
- Dimensionality reductions turns an **intractable** problem into a **tractable** problem
- Dimensionality reduction is extremely useful for **data visualization**

# Drawbacks of Dimensionality Reduction

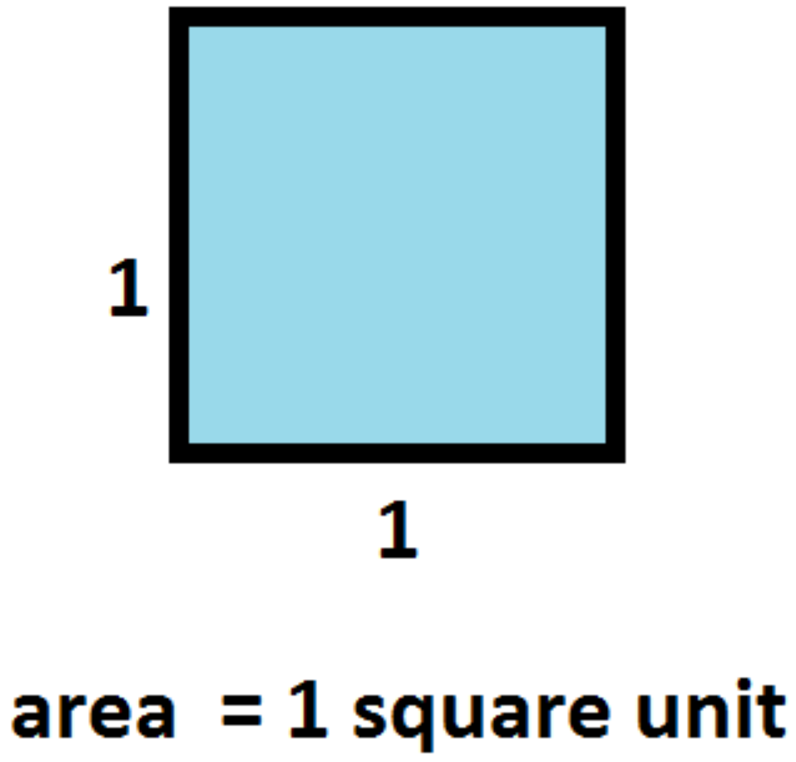
- Through reducing dimensionality, we **lose** some information
- The dimensionality reduction can speed up training, however, lead to a slightly **worse** performance
- In some cases, dimensionality reduction of the training data may filter out some **noise** and unnecessary details
- When dimensionality reduction filter out noise, we get a gain in performance
- This gain of performance is not always given
- Always should firs try to train system on the **original** data

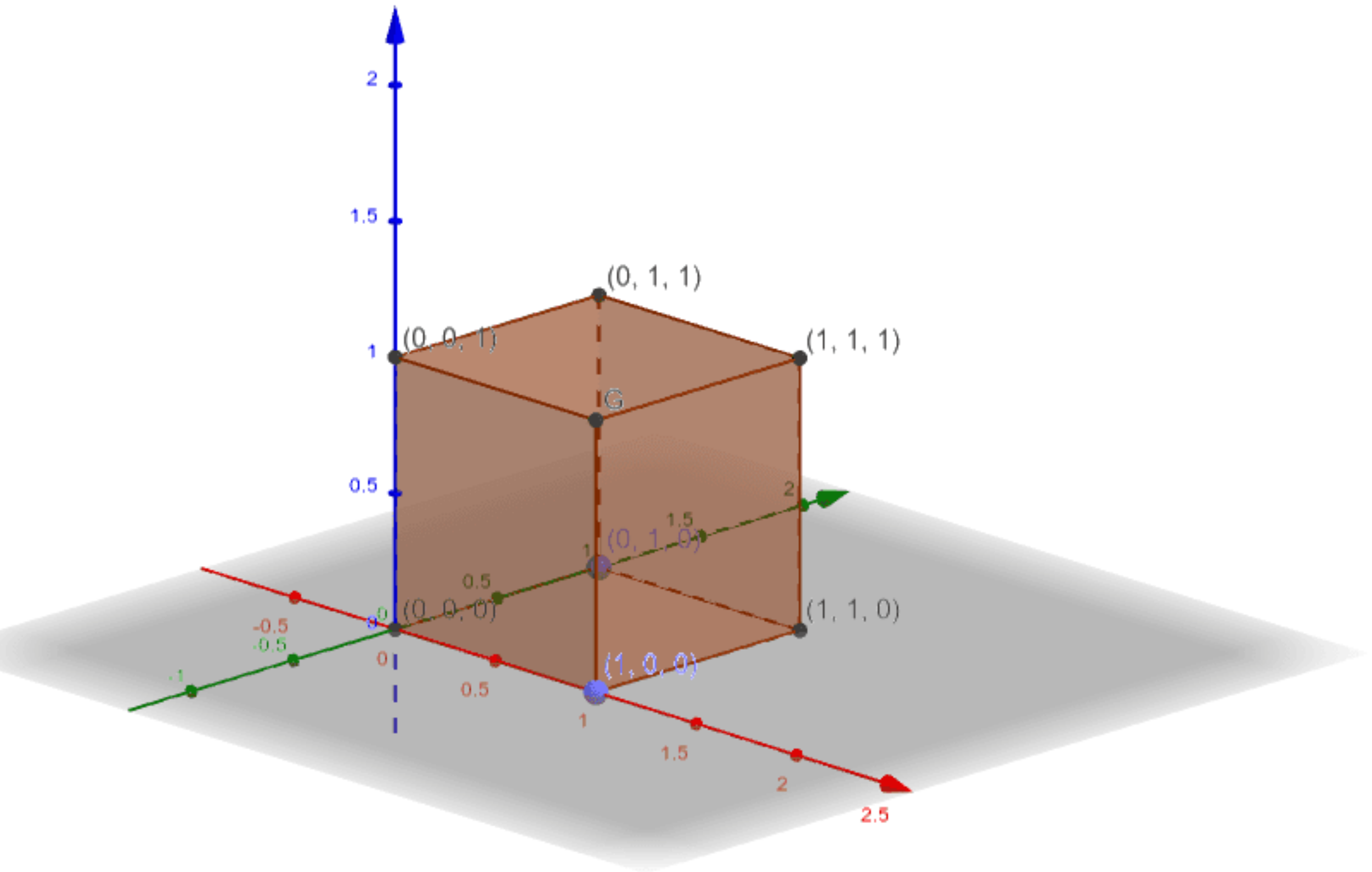
# Curse of Dimensionality

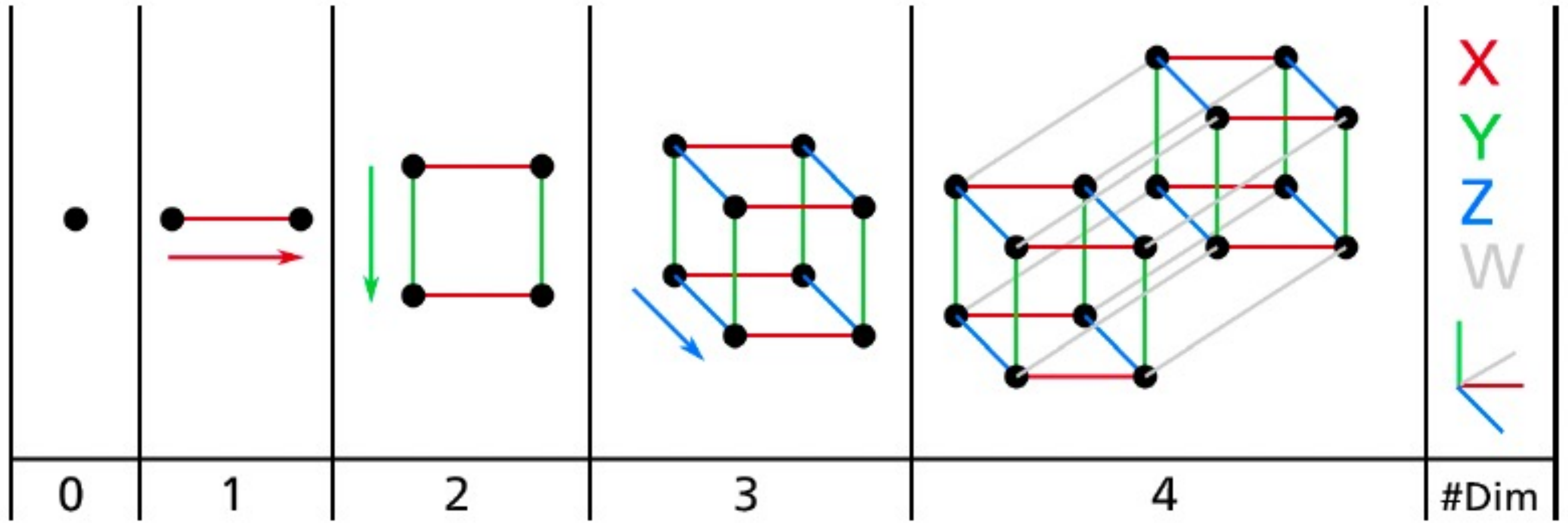
If you pick two points randomly in a **unit square** the distance between these tow points will be, on average, roughly **0.52**

If you pick two points randomly in a **unit 3D cube**, the average distance will be roughly **0.66**

If you pick randomly two points in a **1’000’000-dimensional hypercube**, the average distance will be roughly about **408.25**







# Curse of Dimensionality

High dimensional datasets are very **sparse**

Most training instances are likely to be far **away** from each other

New instance will likely be far away from any training instance

Prediction for new instance will be **less reliable** in high dimension than in lower dimension

More dimensions the training set has, the greater the risk of **overfitting**

One solution to the curse of **dimensionality** cold be to increase the size of the training set to reach a sufficient density of training instances

In practices, it is generally unfeasible to increase the training instances as required

# Main Approaches for Dimensionality Reduction

Two main approaches to reducing dimensionality:

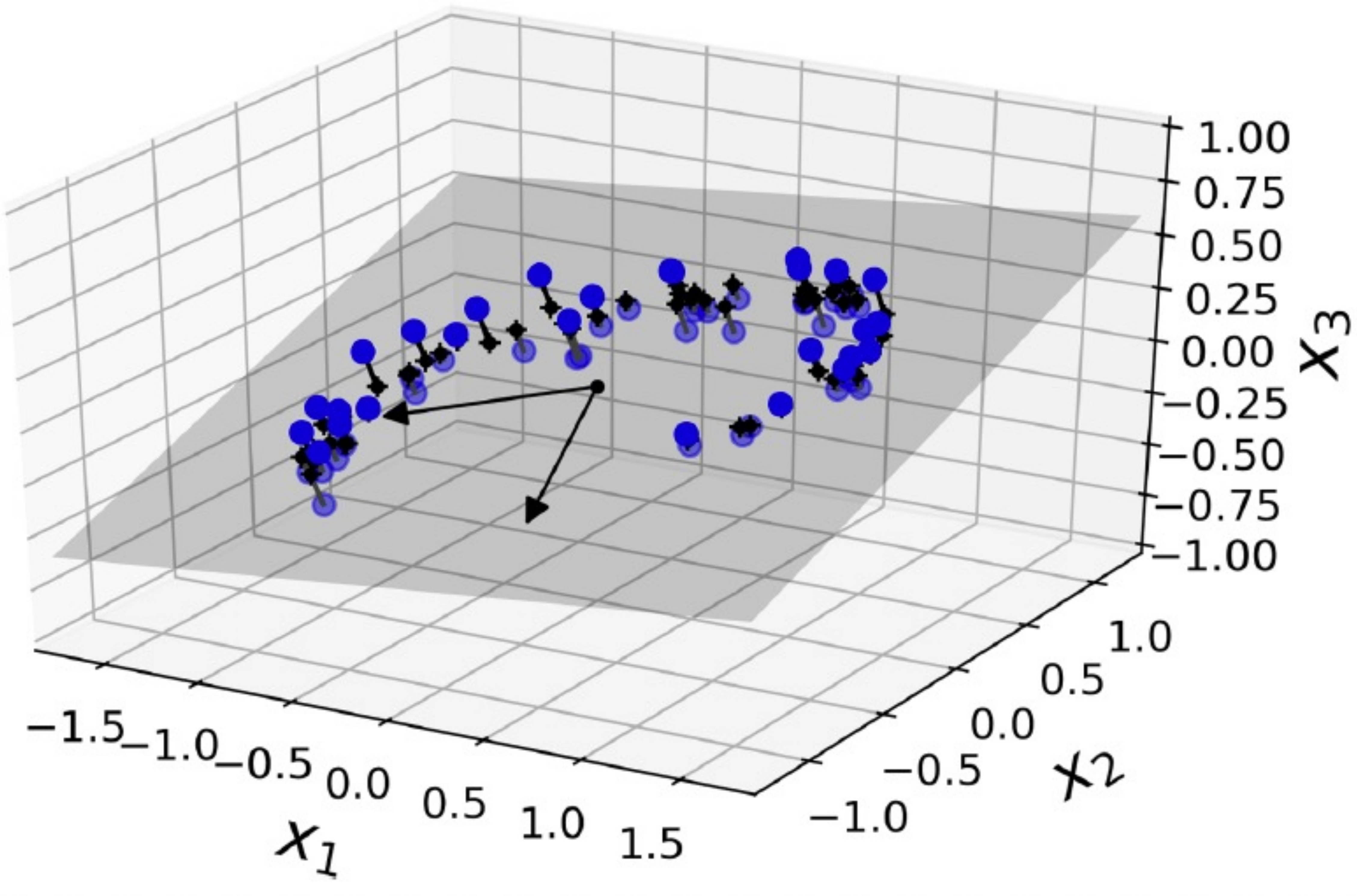
➢ Projection

➢ Manifold Learning

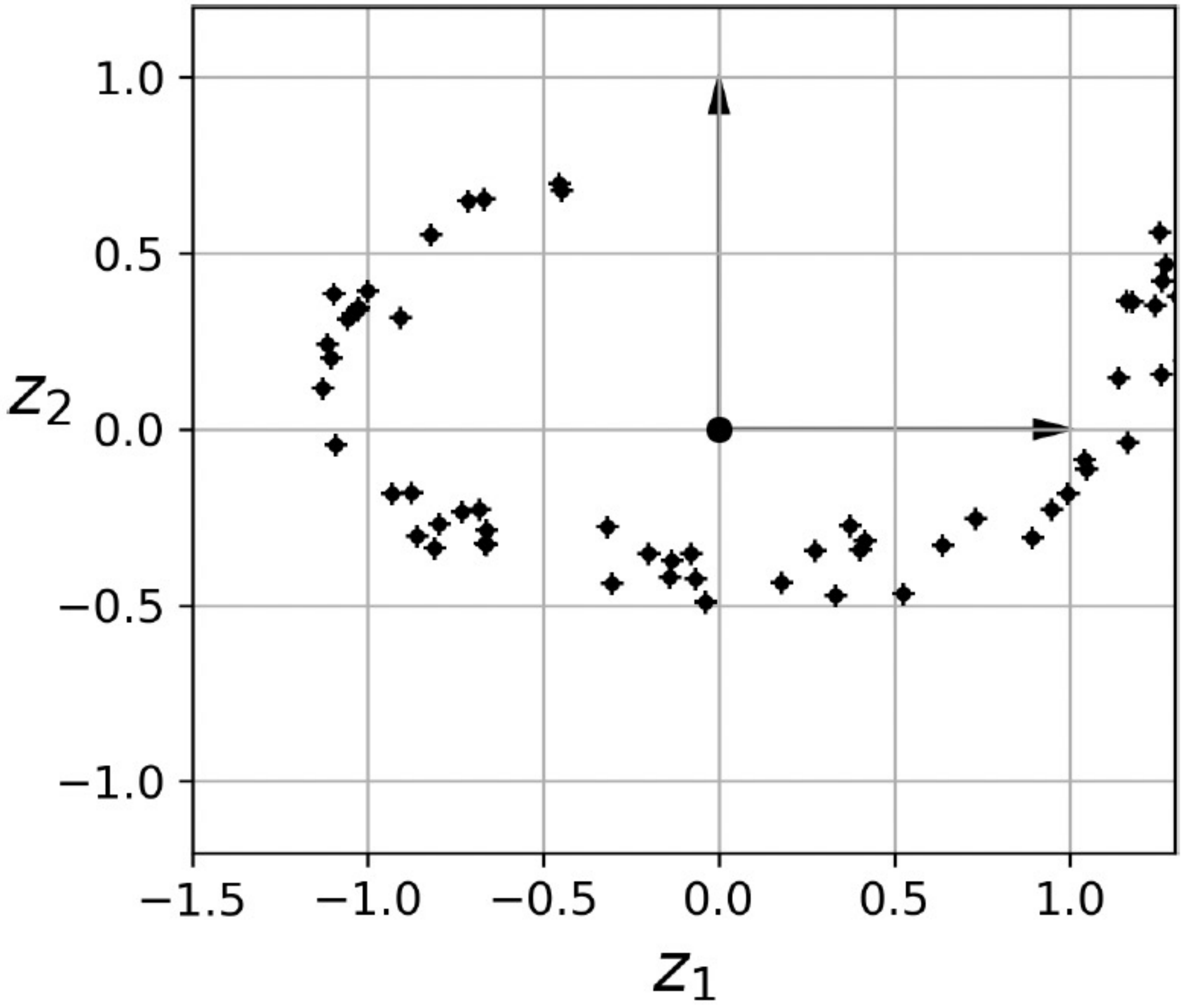
Importance of dimensionality reduction

- In real-world problems, training instances are not spread uniformly across all dimensions
- Many features are constant or highly correlated
- All training instances mostly lie within a much lower-dimensional subspace of the high-dimensional space

# Projection

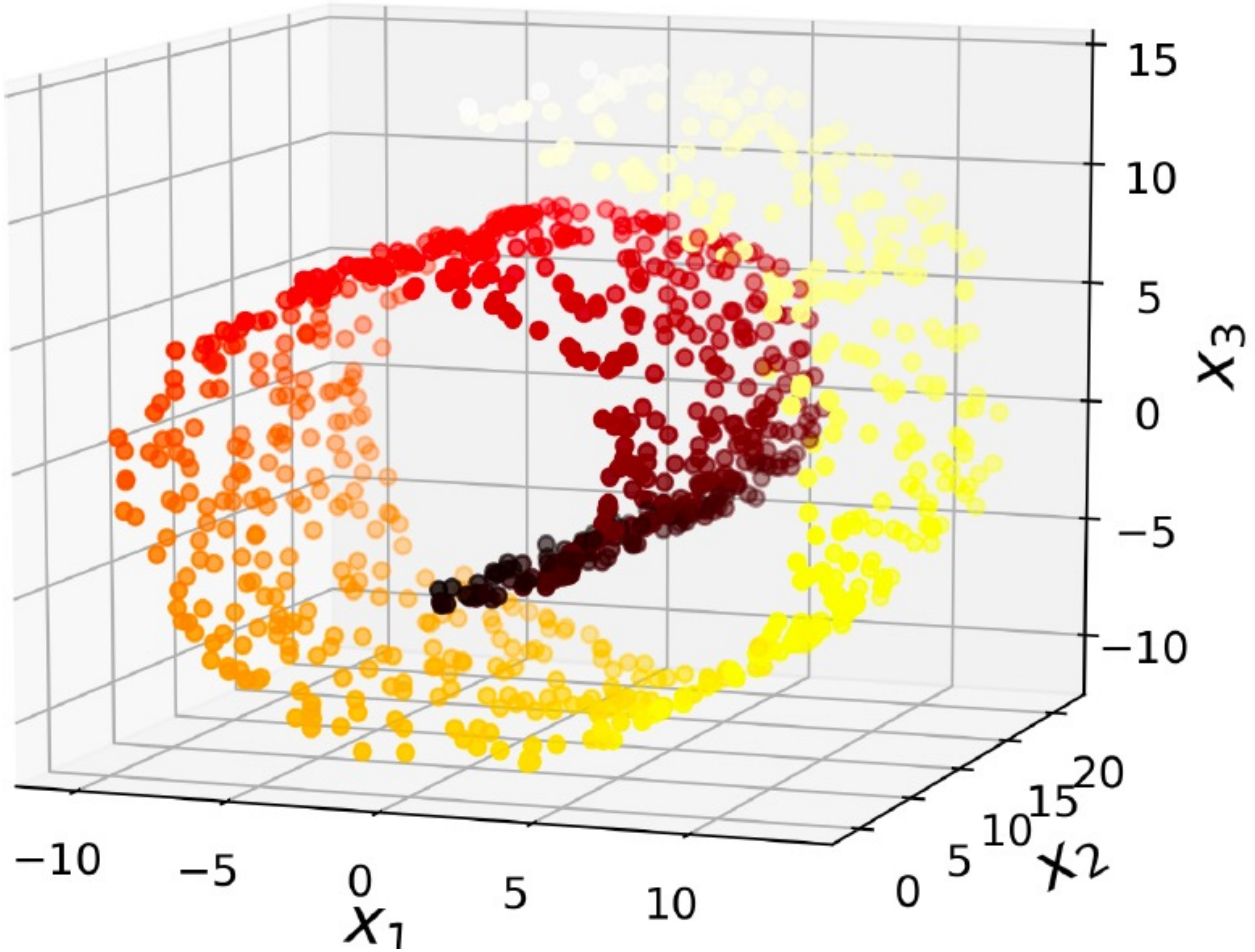


Representation of a 3D dataset (Géron, 2019)



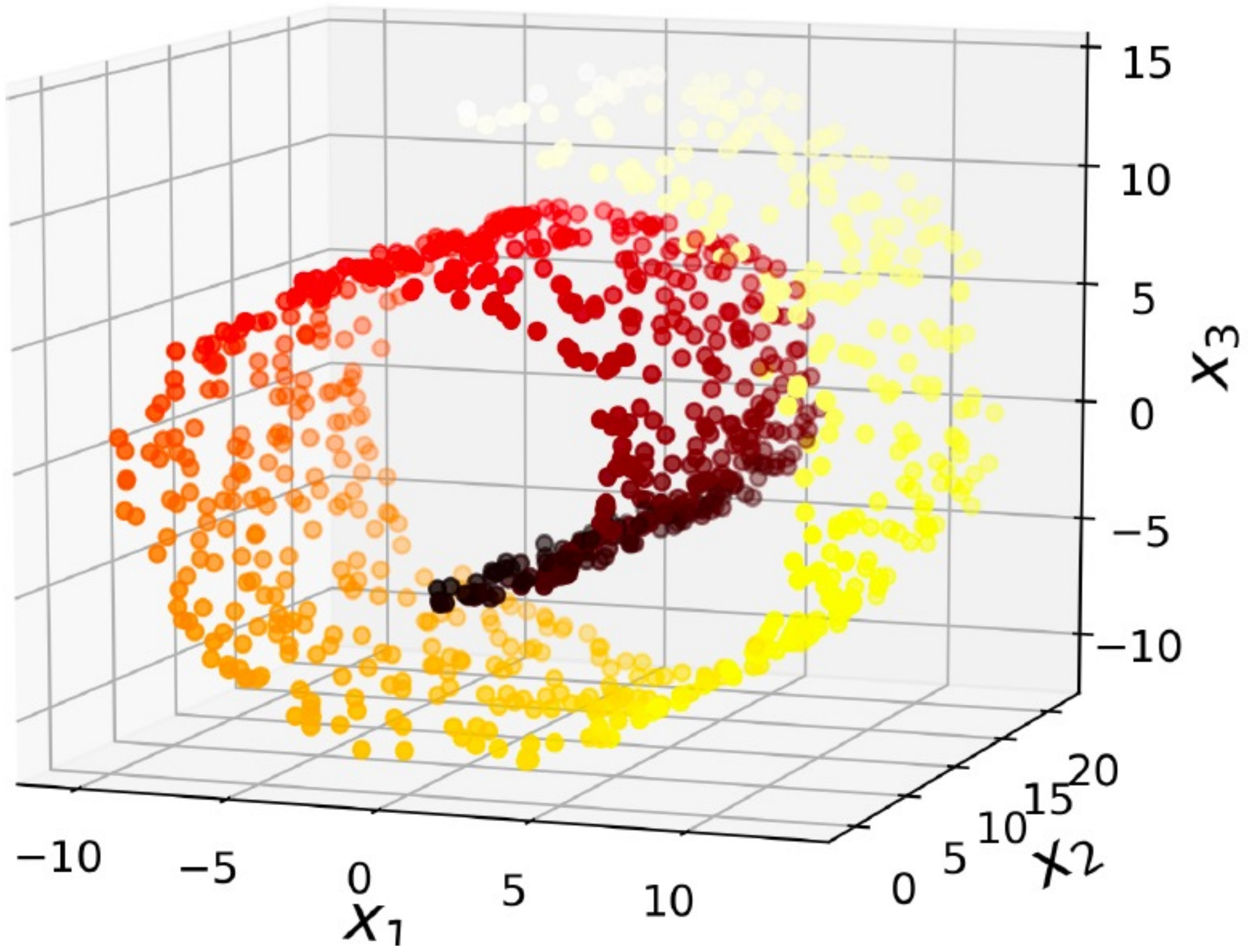
Representation of a 2D project (Géron, 2019)

# Possible Drawbacks of Projection

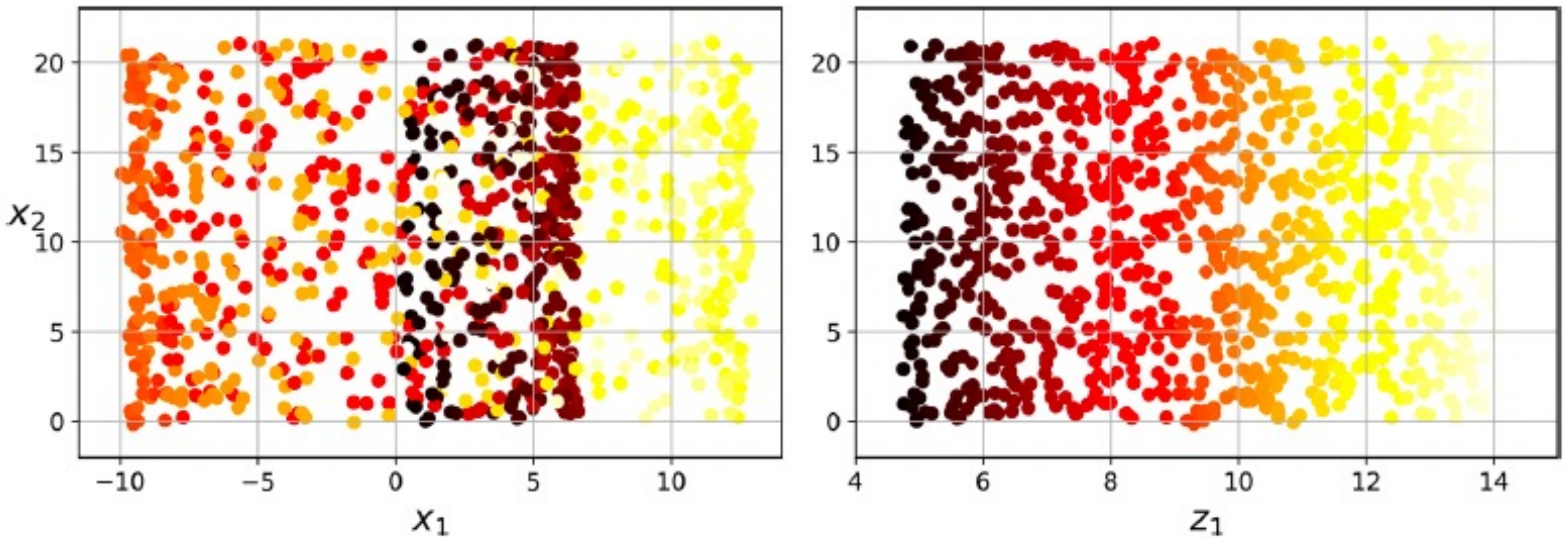


Visualization of the Swiss roll dataset (Géron, 2019)

# Possible Drawbacks of Projection



Visualization of the Swiss roll dataset (Géron, 2019)



Visualization of the Swiss roll dataset: Left projection on the plane, right unrolling Swiss roll (Géron, 2019)

# Manifold Learning

2D manifold is a 2D shape that can be bent and twisted in higher-dimensional space

d-dimensional manifold is a part of a n-dimensional space (where d-dimensional space)

Manifold Laerning reduces the the dimensionality of the raining instances by modleing the manifold on which the trianing instances lies

# Agenda

1. Einführung
2. Kennenlernen Spiel
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. **Principal Components Analysis**
10. T-Distributed Stochastic Neighbor Embedding
11. Zusammenfassung

# Principal Component Analysis

Goal: to find a low representation that captures as much of the information as possible (James et al., 2017)

Let $X_1, X_2, \dots, X_p$ be the p features with n observations.

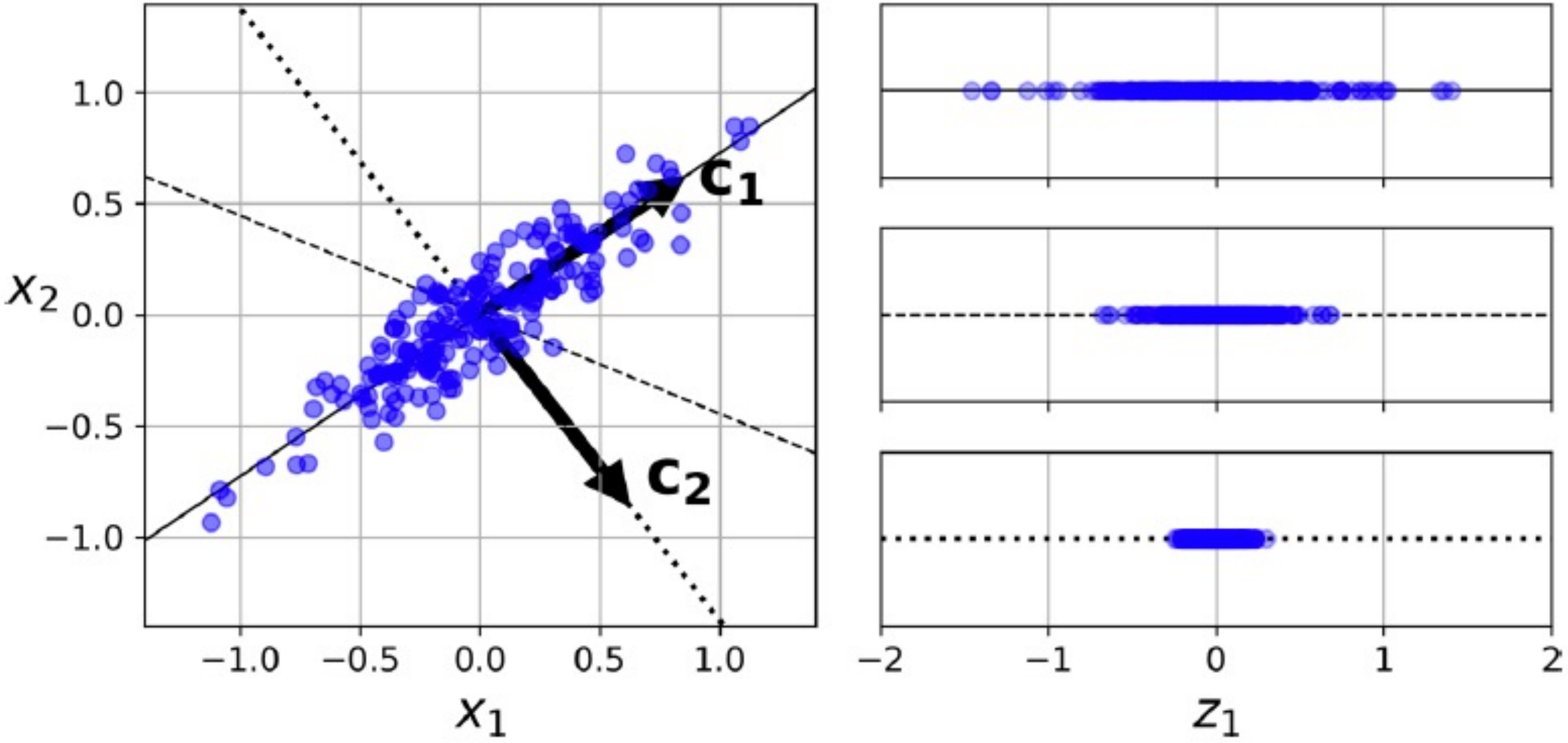
**First** principal component: $Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$

- with largest variance and $\sum_{j=1}^{p} \phi_{j1}^2 = 1$
- $\phi_{j1}$ are the loadings and $z_{i1}$ are the scores.

Optimization Problem for first principal component:

$$\max_{\phi_{11}, \phi_{21}, \dots, \phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left( \sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ with } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

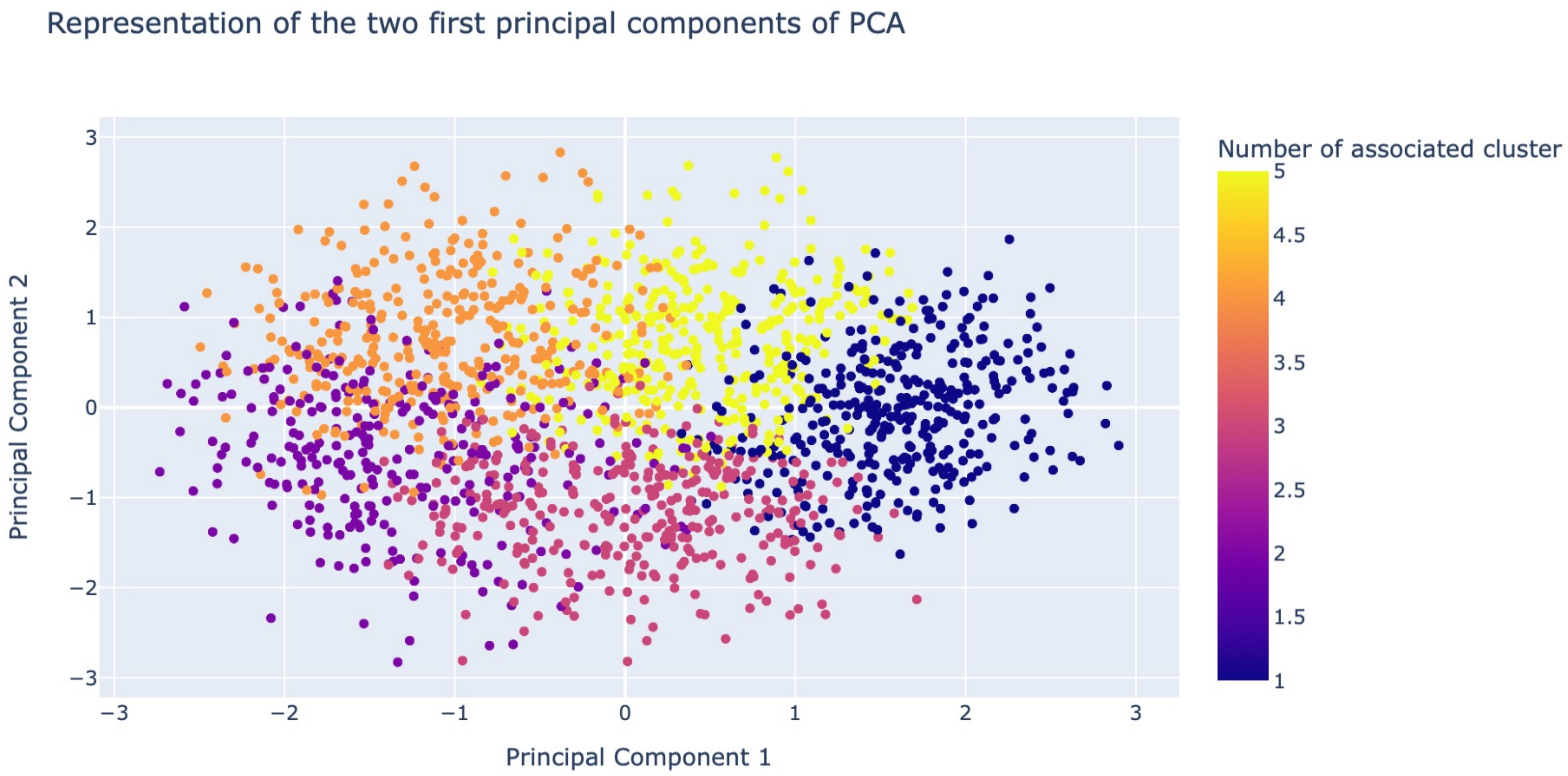
# Selecting the subspace to project on



# Principal Component Analysis

- When the first principal component $Z_1$ of the features has been determined, we can find the second principal component $Z_2$ (James et al., 2017; Kuhn et al., 2016).
- $Z_2$ uncorrelated with $Z_1$

# Example of PCA



# Exercise

- Begeben Sie sich in ein Gruppe von 2 bis 3 Teilnehmenden
- Öffnen Sie das Code Notebook “dimension_reduction_pca_exercise.ipynb”
- Reduzieren Sie den ”Feature” Raum auf zwei Dimensionen
- Dokumentieren Sie ihr Vorgehen und ihre Resultate

# Agenda

1. Einführung
2. Kennenlernen Spiel
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. **T-Distributed Stochastic Neighbor Embedding**
11. Zusammenfassung

# t-Distributed Stochastic Neighbor Embedding (t-SNE)

Video: Design at Large – Laurens van der Maaten, Visualizing Data Using Embeddings
https://www.youtube.com/watch?v=EMD106bB2vY

Video explain t-SNE

Idea of the method is explained from time 11:15 to 24:00

# T-SNE (Stochastic Neighbor Embedding)

Data: data set $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$

Cost function parameters: perplexity (perp)

Optimization Parameters:

- Number of iterations $T$,
- Learning rate $\eta$,
- Momentum $\alpha(t)$

Result: Low-dimensional data representation $\mathcal{Y}^t = \{y_1, y_2, \dots, y_n\}$

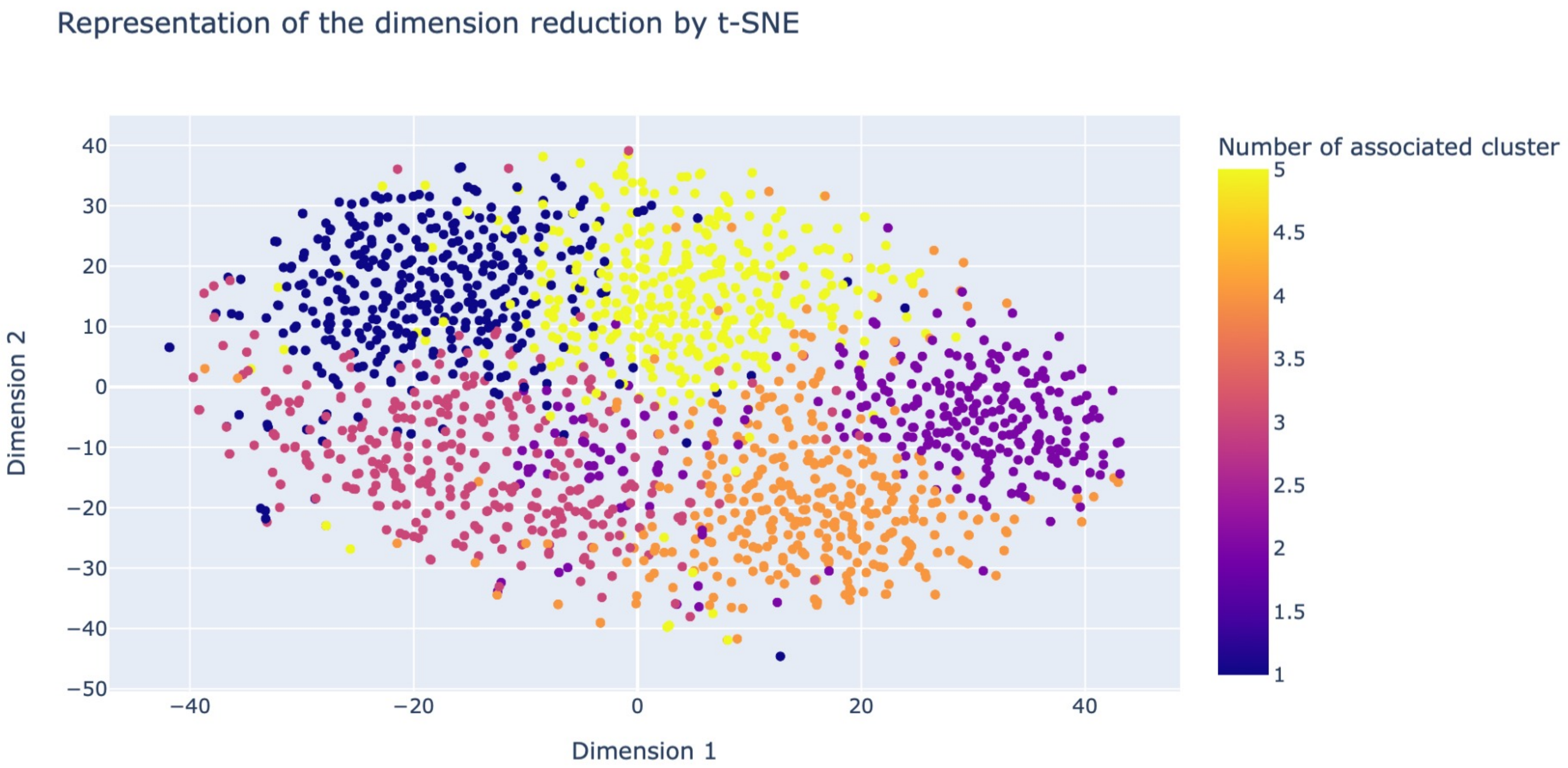
(van der Maaten and Hinton, 2008)

# Pseudo Code T-SNE

```
Begin
        Compute pairwise affinities $p_{j|i} = \frac{exp\left(-\|x_i - x_j\|^2/(2\sigma_i^2)\right)}{\sum_{k\neq i} exp\left(-\|x_i - x_k\|^2/(2\sigma_i^2)\right)}$ with perplexity $Perp(P_i) = 2^{H(P_i)}$
        where $H(P_i)$ is the Shannon entropy $H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$
        Set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2}$
        Sample initial solution $\mathcal{Y}^0 = \{y_1, y_2, \dots, y_n\}$ from $\mathcal{N}(0, 10^{-4}\mathfrak{I})$
        For t = 1 to T do
                Compute low dimensional affinities $q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k\neq l}(1 + \|y_k - y_l\|^2)}$
                Compute gradient $\frac{\partial C}{\partial \mathcal{Y}} = 4\sum_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \|y_i - y_j\|^2\right)^{-1}$
                Set $\mathcal{Y}^t = \mathcal{Y}^{(t-1)} + \eta \frac{\partial C}{\partial \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
        end
End
```

(van der Maaten and Hinton, 2008)

# Example of T-SNE



# Exercise

- Begeben Sie sich in ein Gruppe von 2 bis 3 Teilnehmenden
- Öffnen Sie das Code Notebook “dimension_reduction_tsne_exercise.ipynb”
- Reduzieren Sie den Raum
- Dokumentieren Sie ihr Vorgehen und ihre Resultate

# Agenda

1. Einführung
2. Kennenlernen Spiel
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. **Zusammenfassung**

# Zusammenfassung

Was haben wir gelernt?

# Feedback

Vielen Dank für euer Feedback auf folgender Seite

http://www.evaluationszielscheibe.ch/?disc=7ebb64

# Fragen

Darstellung eines Fragesymbol aufgerufen von der Webseite https://www.qnigge.de/news/detail/modul-v/#images am 12.07.2021.

# Referenzen

Galli, S. (2017). Feature Engineering for Machine Learning, https://github.com/solegalli/feature-engineering-for-machine-learning, Accessed on 15.09.2022.

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn and TensorFlow

Gonick, Larry and Woollcott Smith. 2005. The Cartoon Guide to Statistics. HarperCollins Publishers.

Huang, Z. (1997), Clustering Large Data Sets With Mixed Numeric And Categorical Values, Computer Science.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). An Introduction to statistical learning with Applications in R. Springer.

Kuhn, M. and Johnson, K. (2013). Applied Predictive Modeling. Springer.

Staudt, Y. and Wagner, J. (2022). Assessing the Performance of Random Forests for Modeling Claim Severity in Collision Car Insurance. Risks. Vol 9. No. 53. DOI: 10.3390/risks9030053

Tibshirani, R., Walther, G. and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic.

Wong, K. Y., and Chung, F., 2019; Visualizing Time Series Data with Temporal Matching Based t-SNE.

Yin, S., Gan, G., Valdez, E and Vadiveloo, J. (2021). Applications of Clustering with Mixed Type Data in Life Insurance, arXiv.

HSLU Hochschule Luzern

# Danke!

Datum

FH Zentralschweiz