



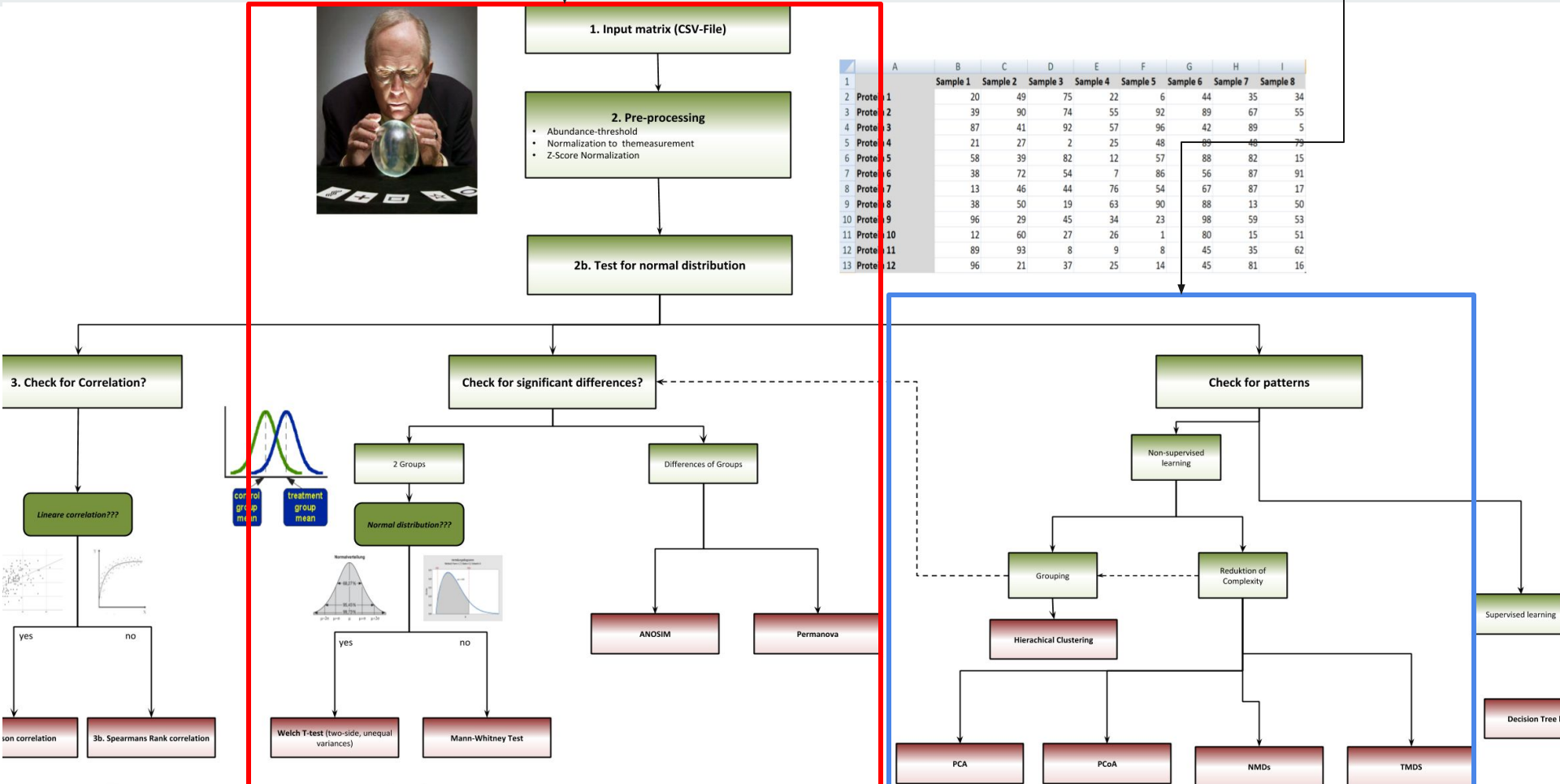
# Multivariate Methods

CLASSIFICATION & ORDINATION

# roadmap

## First Tutorial

## Second Tutorial



	A	B	C	D	E	F	G	H	I
1	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8	
2 Prote	1	20	49	75	22	6	44	35	34
3 Prote	2	39	90	74	55	92	89	67	55
4 Prote	3	87	41	92	57	96	42	89	5
5 Prote	4	21	27	2	25	48	89	48	79
6 Prote	5	58	39	82	12	57	88	82	15
7 Prote	6	38	72	54	7	86	56	87	91
8 Prote	7	13	46	44	76	54	67	87	17
9 Prote	8	38	50	19	63	90	88	13	50
10 Prote	9	96	29	45	34	23	98	59	53
11 Prote	10	12	60	27	26	1	80	15	51
12 Prote	11	89	93	8	9	8	45	35	62
13 Prote	12	96	21	37	25	14	45	81	16

---

## GOALS :

Overview : Normalization & Group Comparison

Moving Ahead - Multivariate Methods : Supervised Learning & Unsupervised Learning, Ordination & Clustering

Ordination: PCA, PCoA & NMDS

Grouping: Clustering

---

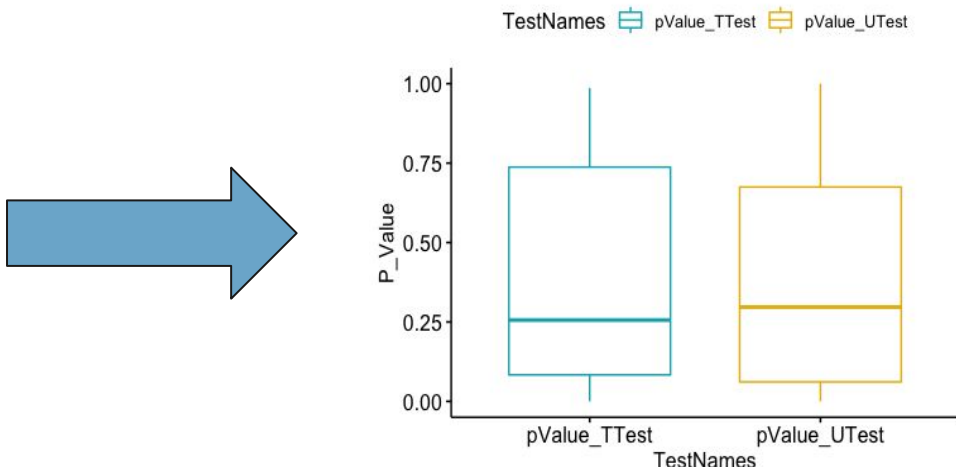
# OVERVIEW

- what are we upto?
- keywords

***NORMALIZATION? GROUP  
COMPARISON? WHY ALL THE  
FUSS?***

- previously unclarified  
**p-value, w-value, Bonferoni  
Correction T-Test, Benjamin  
Hochberger Correction T-Test**

# what are we upto?



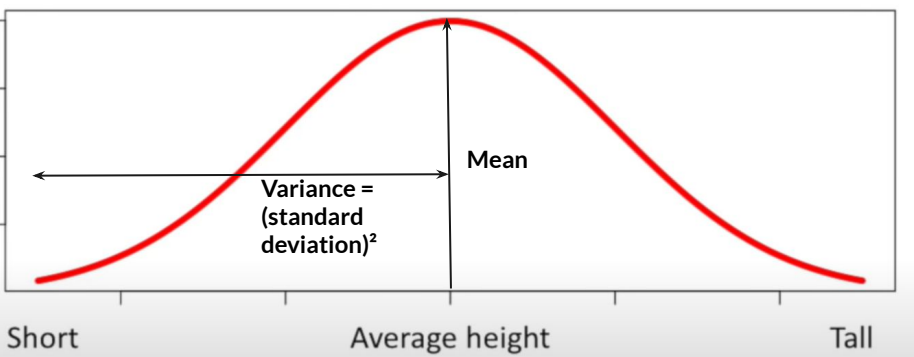
to **visualize** large amounts of complex **data** is easier than poring over spreadsheets or reports. ... **Data visualization** can also: **Identify areas that need attention or improvement.**

## how?

### Statistical Tools through R :

- Normalization
- Group Comparison (**T-Test, PERMANOVA etc.**)
- Multivariate Methods (**Clustering, Ordination**)

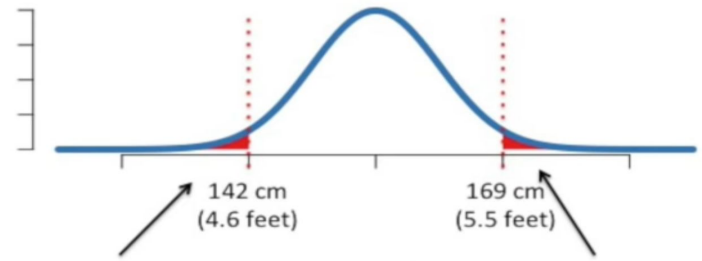
# keyword : normalization



## p-value

To calculate p-values, you add up the percentages of areas under the curve.

For example, the p-value for someone who is 142 cm tall is...

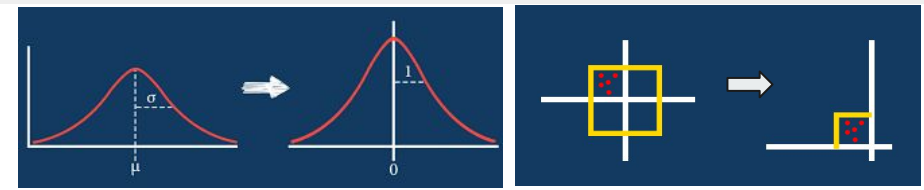


The 2.5% of the area for people 142 cm or shorter. + The 2.5% of the area for people 169 cm or taller. = 0.05

## why bother?

- Robust **visualization** of a data or data variable - possible to create null hypothesis and test them
- **data normalization** when seeking for **relations**
- as part of data preparation for **machine learning**. The goal of **normalization** is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values
- Easy to **compare** data or data variables

## how?

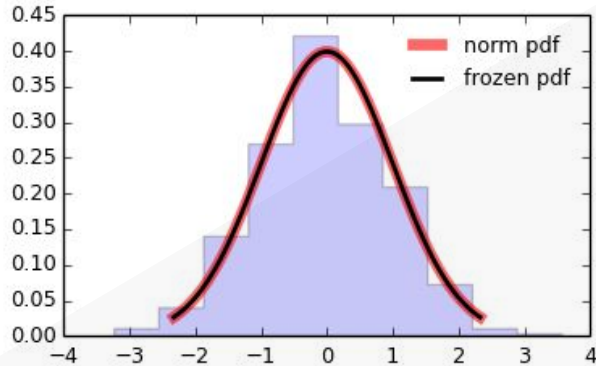


<u>Name(ID)</u>	<u>Age</u>	<u>Height</u>	<u>Gender</u> (1=f, 2=m, 3=other)	<u>Education Level</u> (0=Bachelor, 1= Master, 2= Post Doc)	<u>Class Label : Teacher(1) or Student(0)</u>
Robert	30	6.1	m(2)	Post Doc(2)	Teacher(1)
Julian	26	6.3	m(2)	Master(1)	Student(0)
Danial	25	5.8	m(2)	Master(1)	Student(0)
Max	26	5.9	m(2)	Master(1)	Student(0)
Faizan	23	6.0	m(2)	Master(1)	Student(0)
Abdullah	27	5.8	m(2)	Master(1)	Student(0)
Ammar	26	5.9	m(2)	Master(1)	Student(0)
Rahul	25	5.8	m(2)	Master(1)	Student(0)
<b><u>Mean</u></b>	<b>26</b>	<b>5.95</b>	<b>2</b>	<b>1.125</b>	

<u>Name(ID)</u>	<u>Age</u>		<u>Height</u>		<u>Gender</u> (1=f, 2=m, 3=other)	<u>Education Level</u> (0=Bachelor, 1= Master, 2= Post Doc)	<u>Class Label : Teacher(1) or Student(0)</u>
Robert	30	1	6.1	3/5	m(2)	Post Doc(2)	Teacher(1)
Julian	26	3/7	6.3	1	m(2)	Master(1)	Student(0)
Danial	25	2/7	5.8	0	m(2)	Master(1)	Student(0)
Max	26	3/7	5.9	1/5	m(2)	Master(1)	Student(0)
Faizan	23	0	6.0	2/5	m(2)	Master(1)	Student(0)
Abdullah	27	4/7	5.8	0	m(2)	Master(1)	Student(0)
Ammar	26	3/7	5.9	1/5	m(2)	Master(1)	Student(0)
Rahul	25	2/7	5.8	0	m(2)	Master(1)	Student(0)
<u>Mean</u>	26	3/7	5.95	0.3	2	1.125	



## Test for Normality: Shapiro-Wilk Test



```
> shapiro.test(matrix$BE_03)
```

Shapiro-Wilk normality test

data: matrix\$BE\_03

W = 0.38432, p-value = 1.103e-14

### Assumption Checks ▼

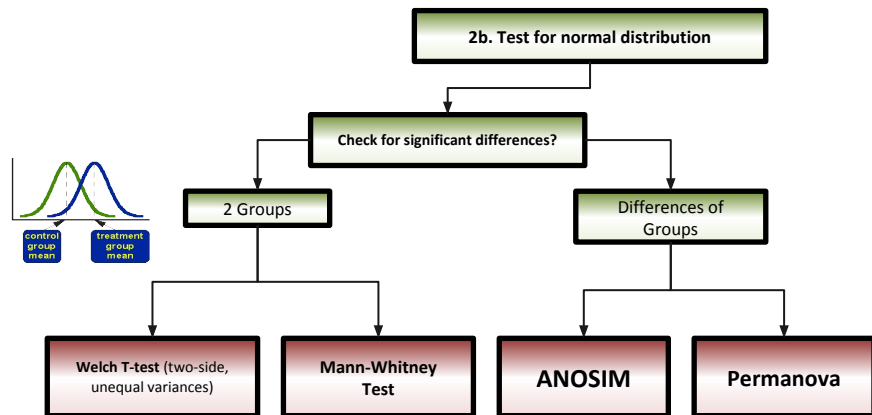
#### Test of Normality (Shapiro-Wilk) ▼

	W	p
Difference	0.938	0.325

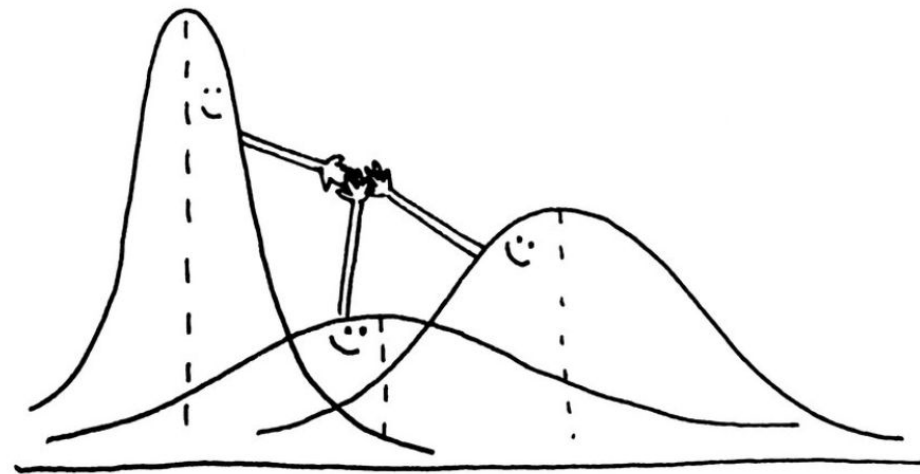
Note. Significant results suggest a deviation from normality.

- Using w-value, we create a NULL hypothesis
  - *if W is very small then the distribution is probably not normally distributed*
- If  $P < 0.05$ , we reject the NULL Hypothesis

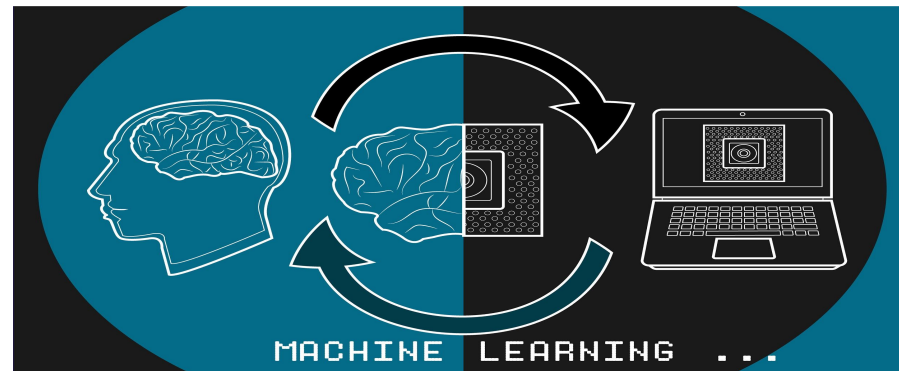
# So now that we have data(normalized), what next?



- check for **Significant Differences (Group Comparison)**
  - between 2 or more groups
    - T-Test & U-Test
    - ANOSIM & PERMANOVA
    - ANOVA & Kruskal-Wallis Test
- infer **Knowledge** out of dataset and/or **prove hypothesis**



and why is this important?



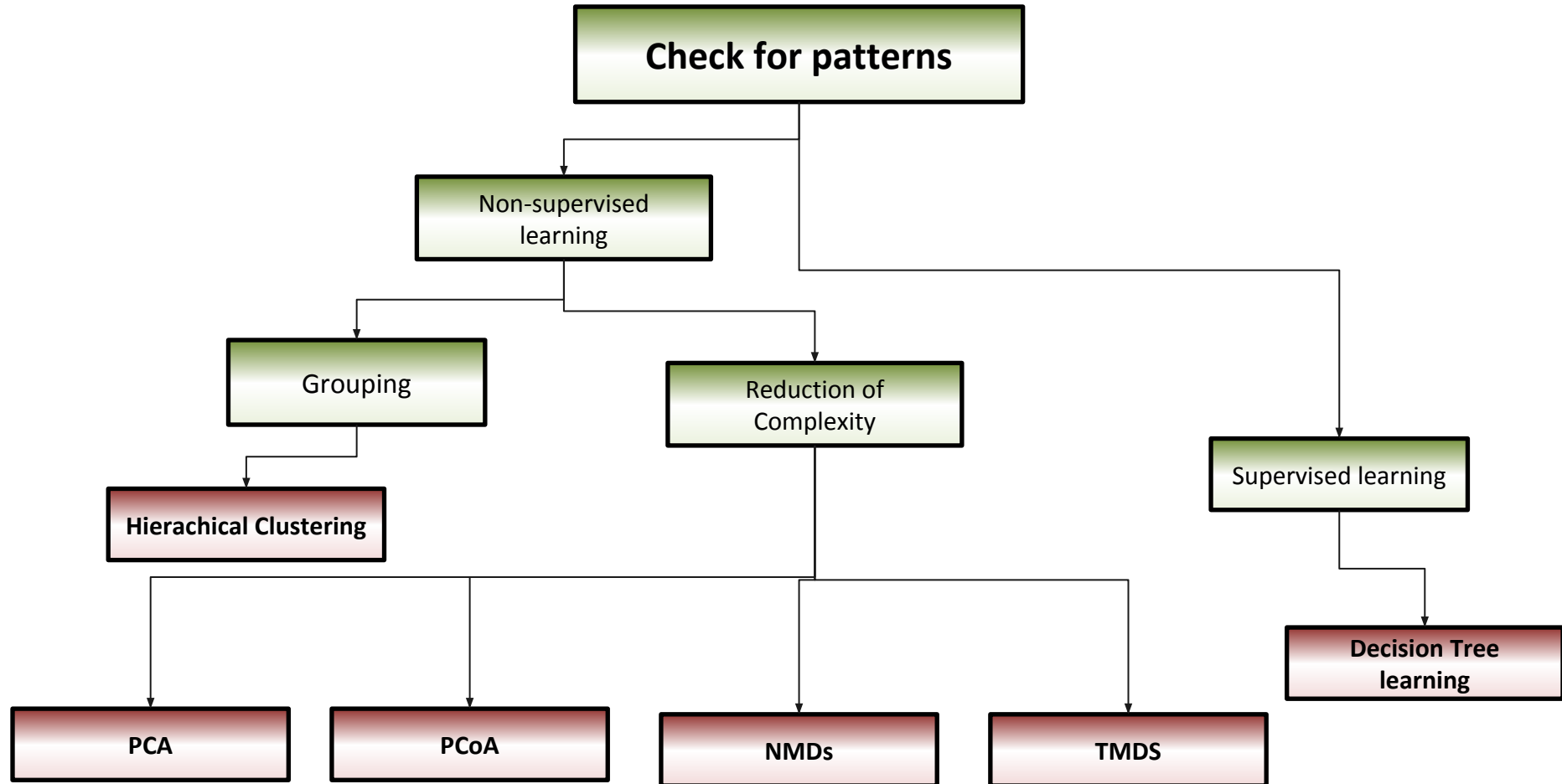
**can DATA be NOT Normalized & still  
make sense??**

—

---

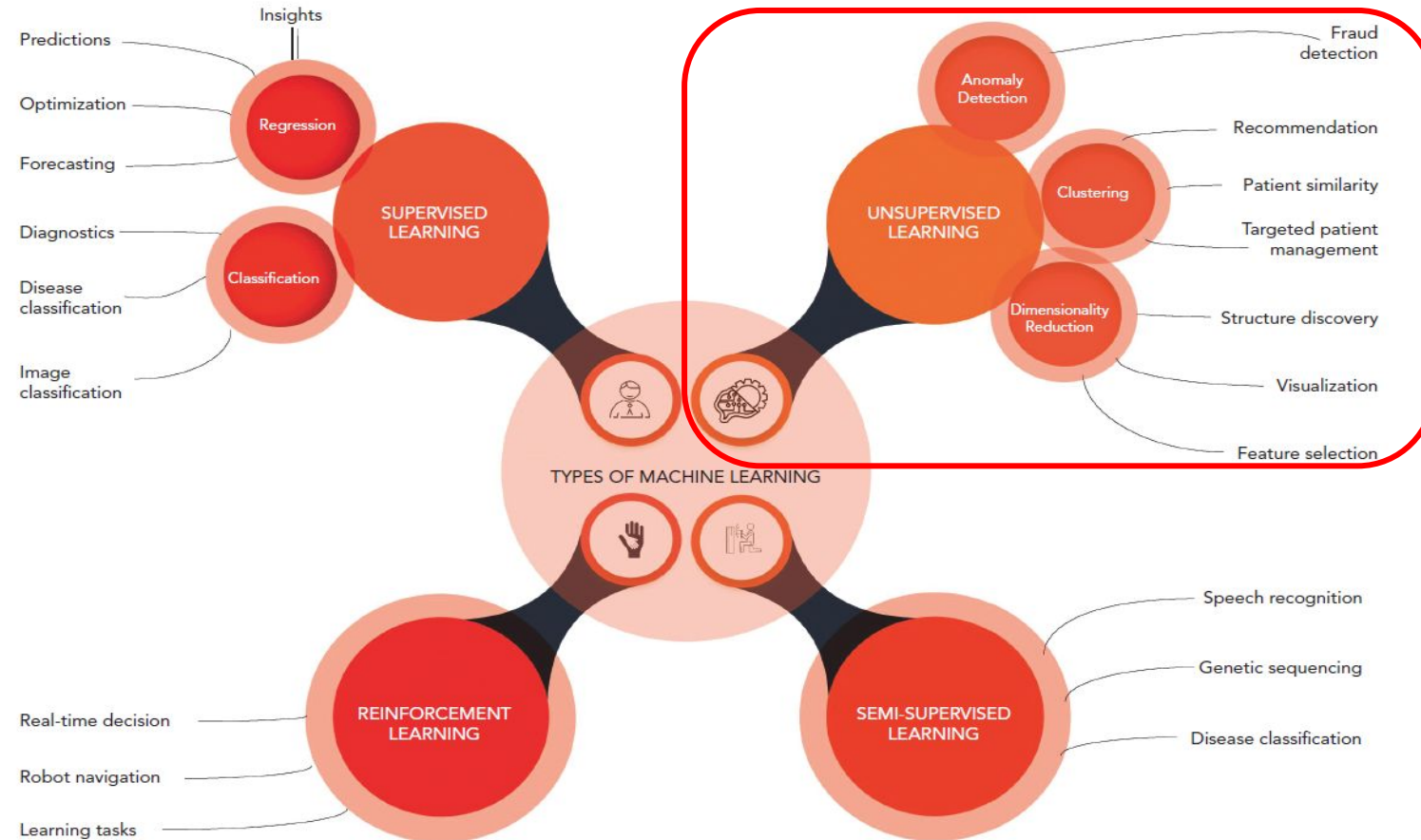
# Multivariate Methods : Ordination & Classification

- unsupervised learning vs supervised learning
- Ordination
  - Grouping
    - Clustering
  - Dimension/Complexity Reduction
    - PCA
    - PCoA
    - NMDS
    - CCA



# types of Machine Learning

Figure 1: Types of Machine Learning with Examples of Respective Use



## supervised learning

Input data



Annotations

These are apples



Model



Prediction

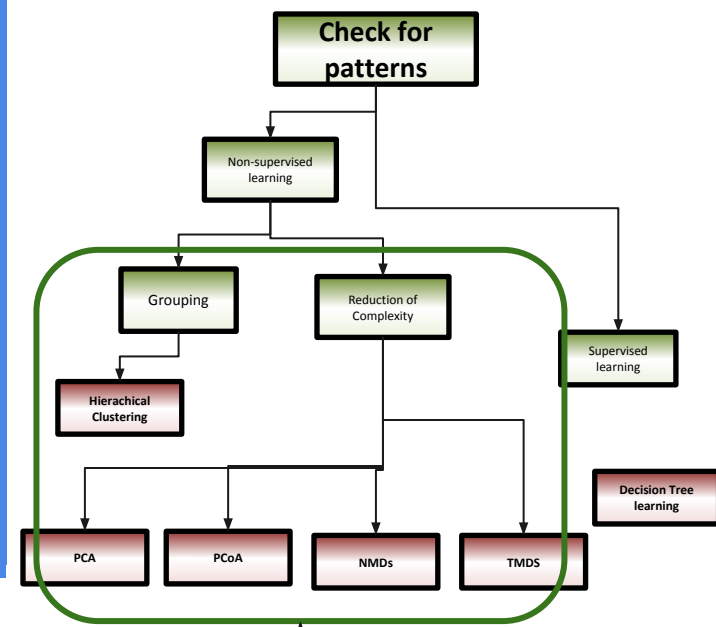
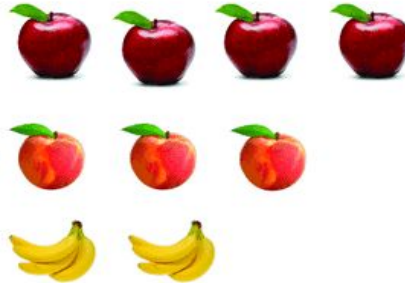
Its an apple!

## unsupervised learning

Input data



Model



**ORDINATION**



**unsupervised learning** vs/ & **supervised learning**

**what is DATA to a Machine??**

---



# unsupervised learning

- grouping

- Clustering

to find **Similarities & Recommendations**

- reduction of Dimension and/or Complexity

- Principal Component Analysis (**PCA**)
- Principal Coordinate Analysis (**PCoA**)
- Non Metric MultiDimensional Scaling (**NMDS**)
- Canonical Correspondence Analysis (**CCA**)

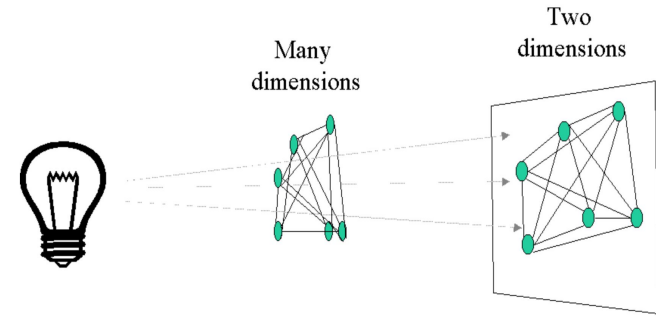
**Structure Discovery, Feature Selection & Visualization**

why?

how?

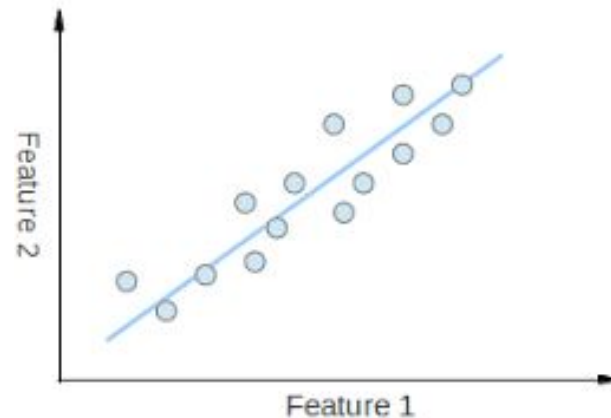
## ordination (an unsupervised approach)

**Ordination** is a collective term for multivariate techniques which summarize a **multidimensional dataset** in such a way that when it is projected onto a **low dimensional space**, any intrinsic pattern the data may possess becomes apparent upon visual inspection.



**why?**

Ordination can be used on the analysis of any set of multivariate objects.



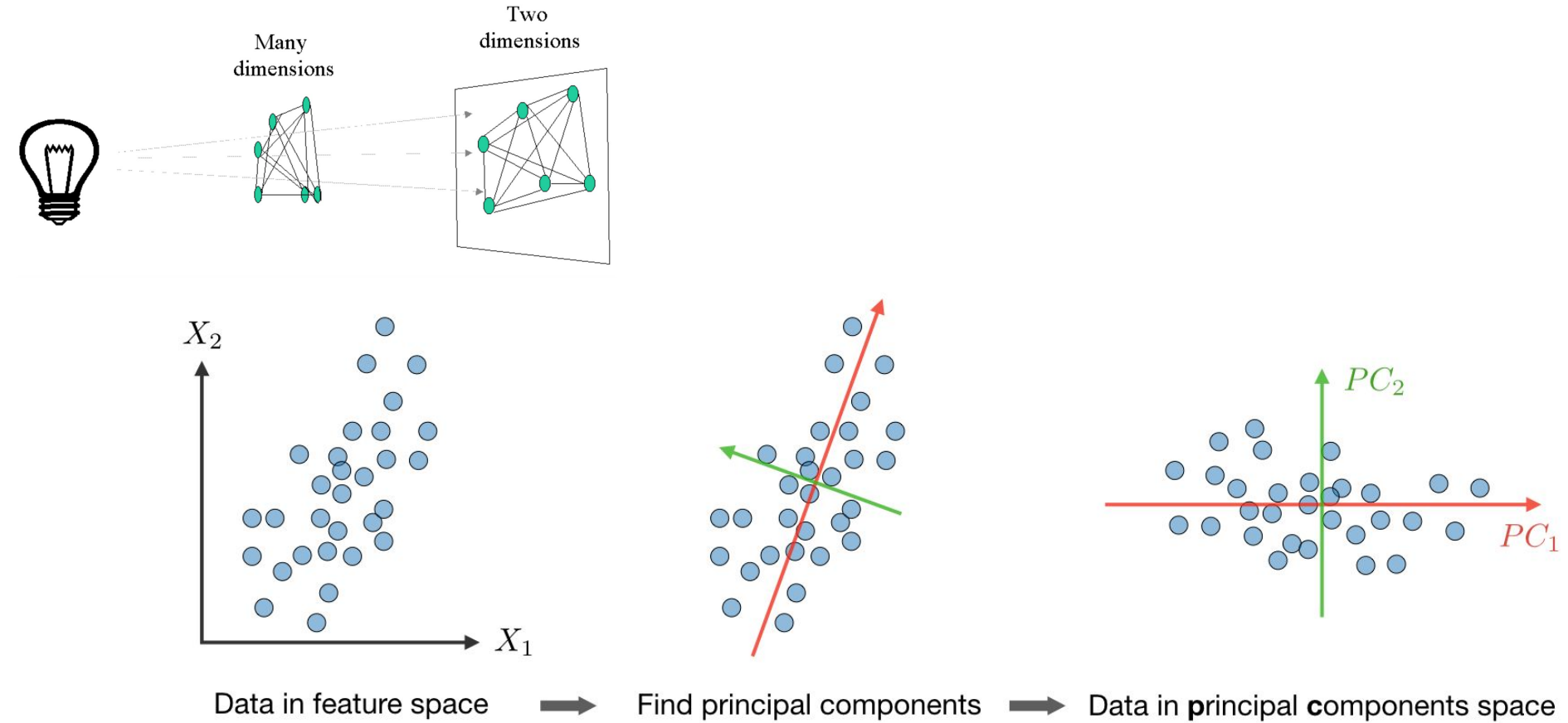
**how?**

---

# Ordination

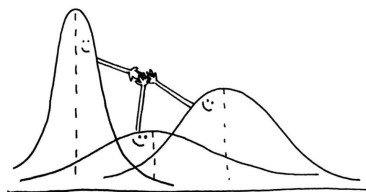
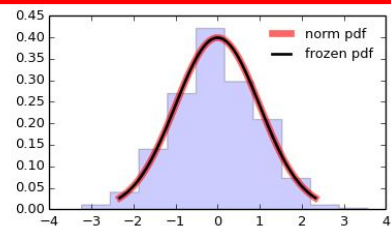
- Dimension Reduction
  - **PCA** (Principal Component Analysis)
  - **PCoA** (Principal Coordinates Analysis)
  - **NMDS** (Non metric Multidimensional Scaling)

# PCA (Principal Component Analysis)



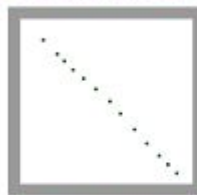
# Steps (PCA)

## 1. Normalize the Dataset



## 2. Compute Covariance Matrix

COVARIANCE



Large Negative Covariance

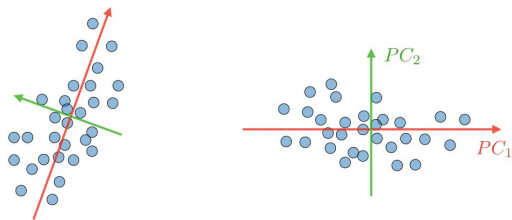


Near Zero Covariance



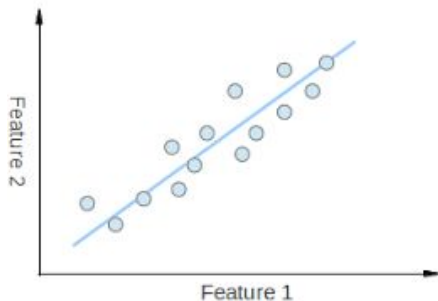
Large Positive Covariance

## 4. Compute Transformation

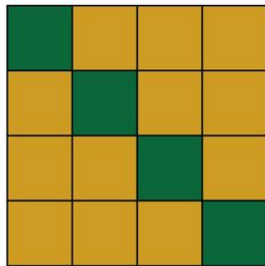


Find principal components → Data in principal components space

## 4. Determine Principal Component



1	2	0	1
-1	7	3	0
5	1	2	9
2	4	5	1



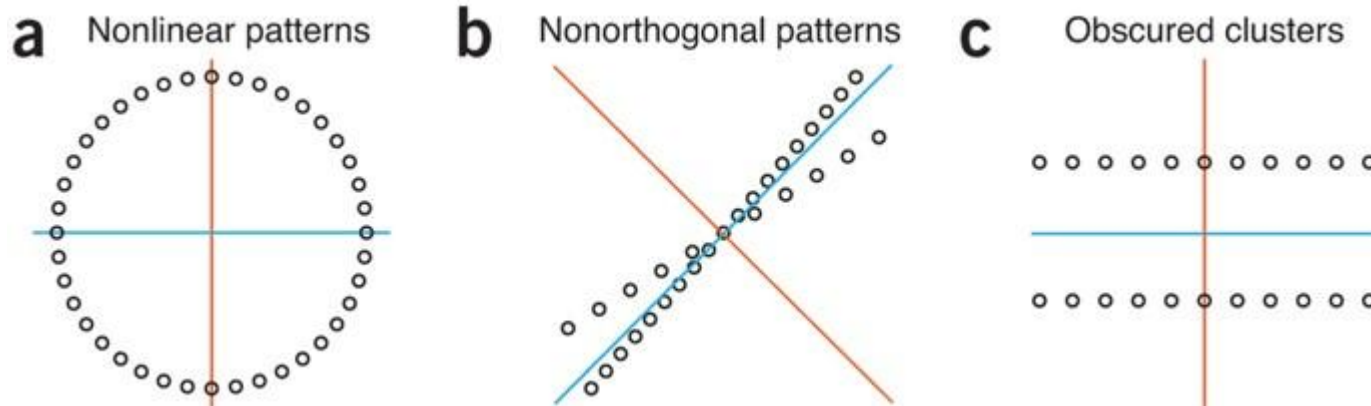
## 3. Perform Eigen Decomposition

## 6. VISUALIZATION

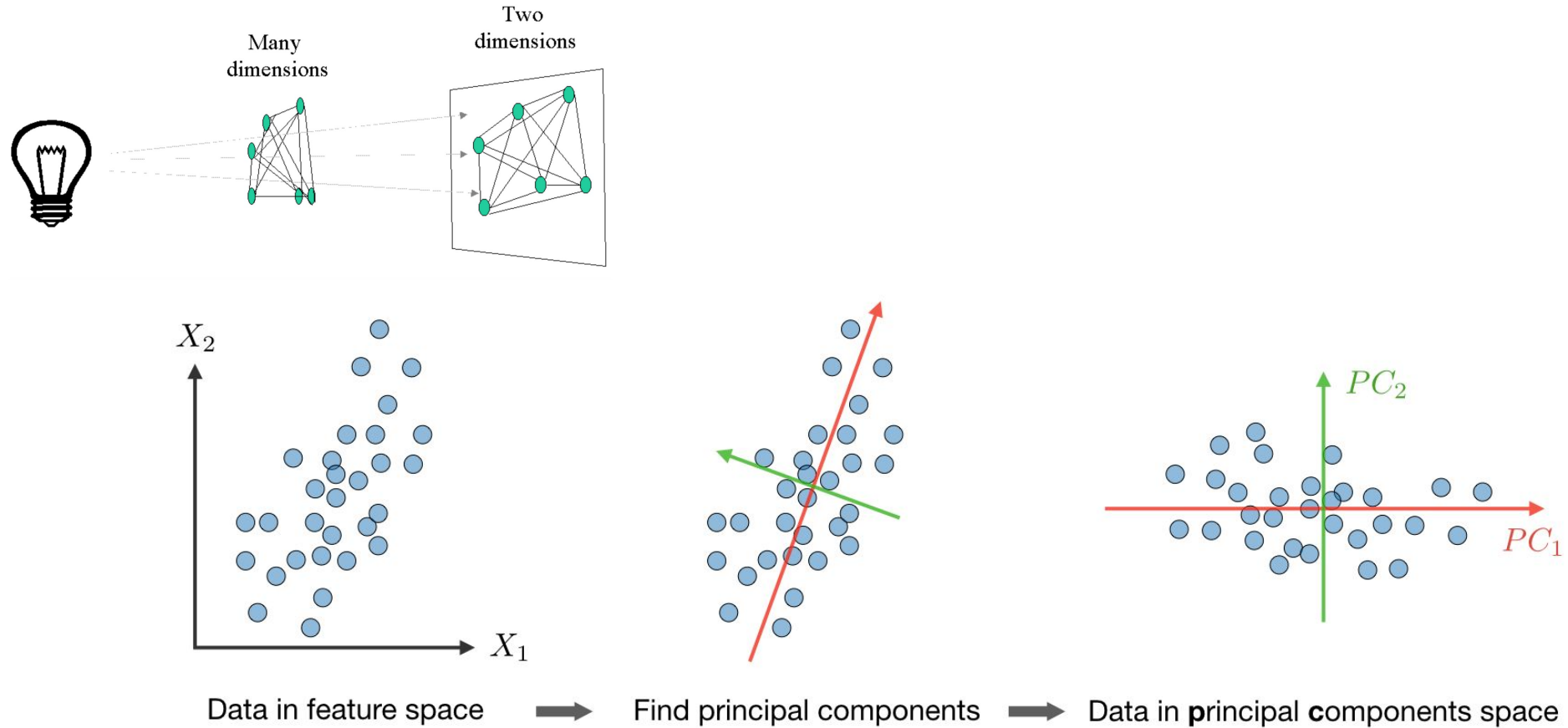
## importance(PCA)

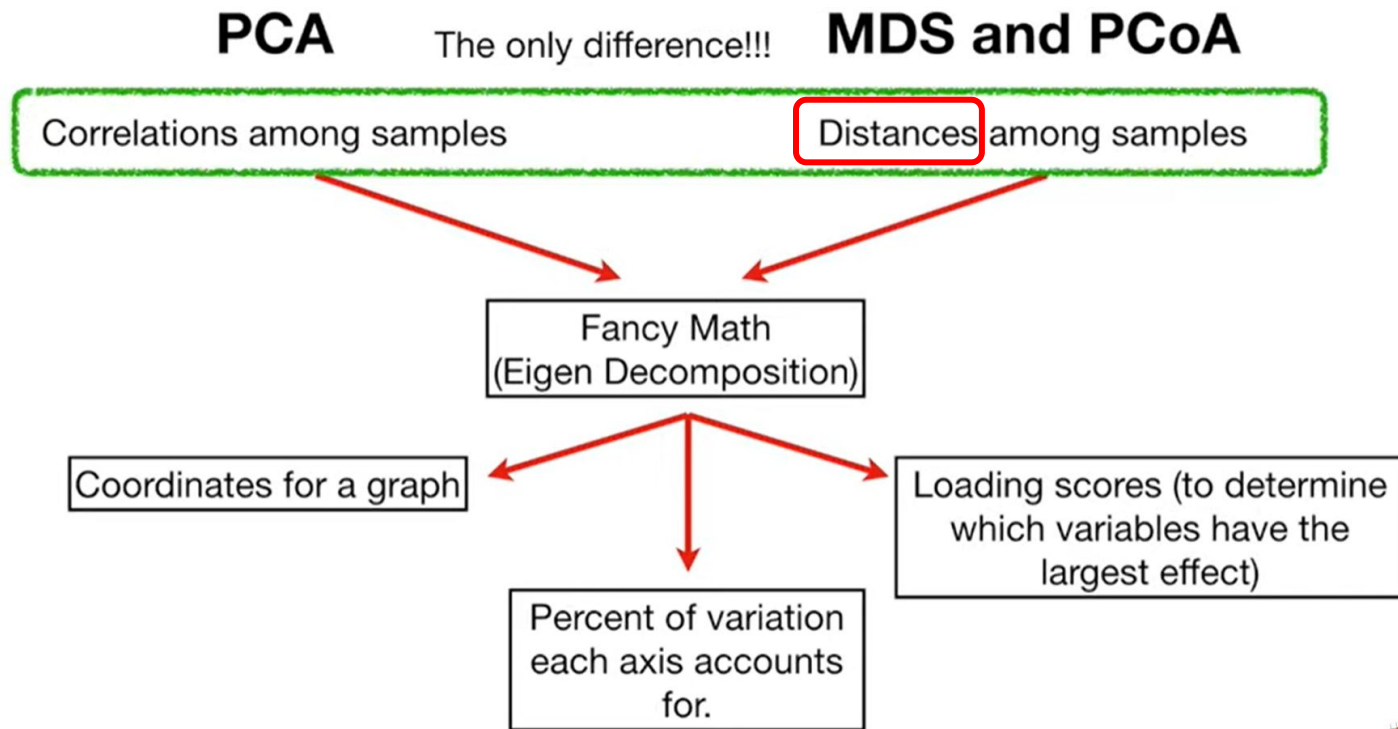
PCA helps you discover correlations & interpret your data, but it will not always find the important patterns.

Principal component analysis (PCA) **simplifies the complexity in high-dimensional data while retaining trends and patterns.** It does this by transforming the data into fewer dimensions, which act as summaries of features



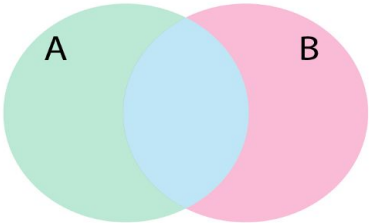
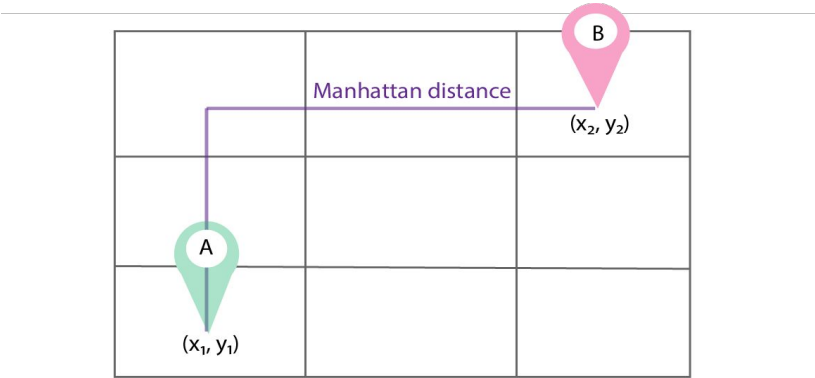
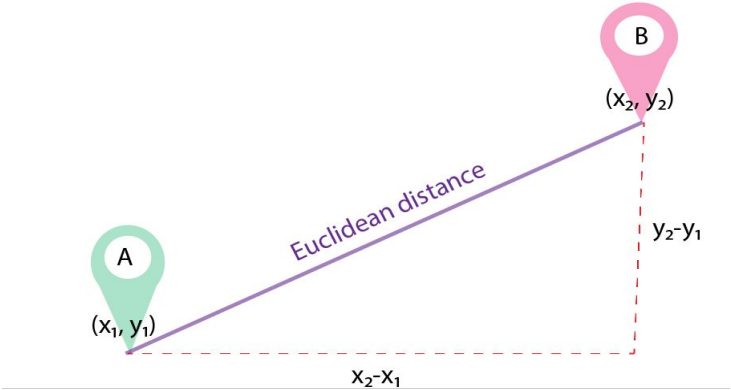
# PCoA (Principal Component Analysis)/ metric multidimensional scaling



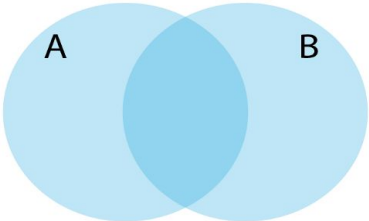




# Distance/ Proximity Measures

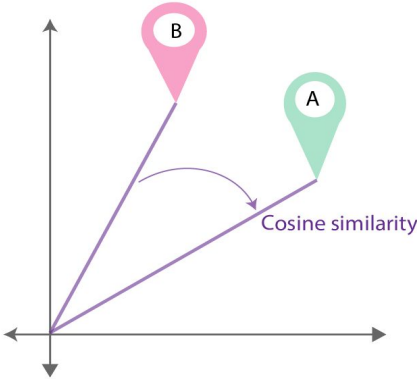


Intersection



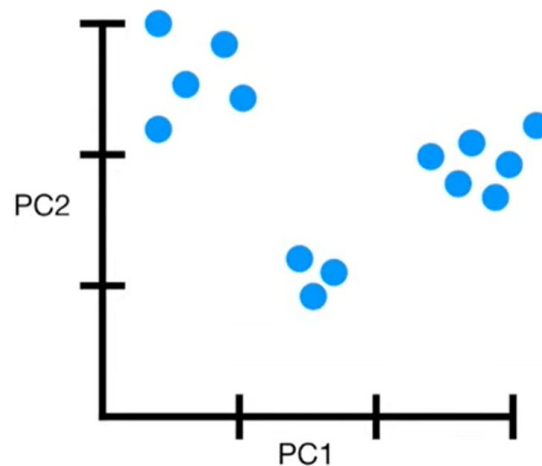
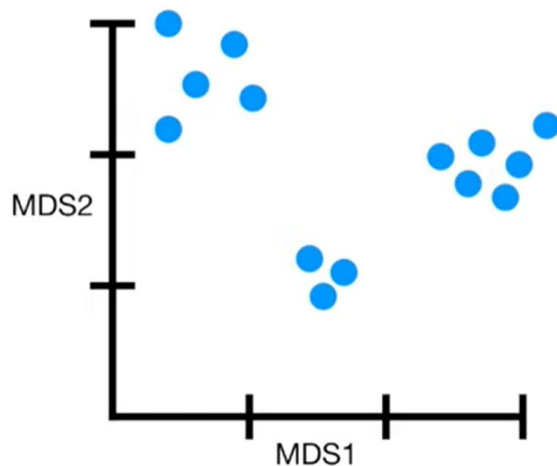
Union

Jaccard Distance



IF we use Euclidean Distance in PCoA, the graph would be similar to a PCA graph

In other words, clustering based on  
**minimizing the linear distances is**  
**the same as maximizing the linear**  
**correlations.**



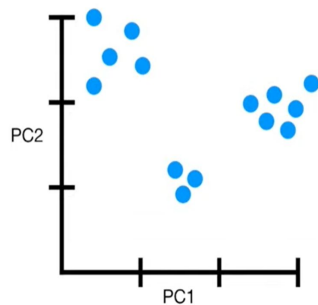
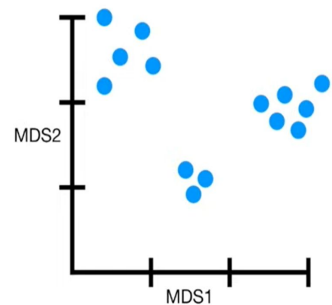
**As with other ordination techniques such as PCA and CA, PCoA produces a set of uncorrelated (orthogonal) axes to summarise the variability in the data set.**

While PCoA is suited to handling a wide range of data, information concerning the original variables cannot be recovered.

# How do I interpret a PCA/PCoA plot?

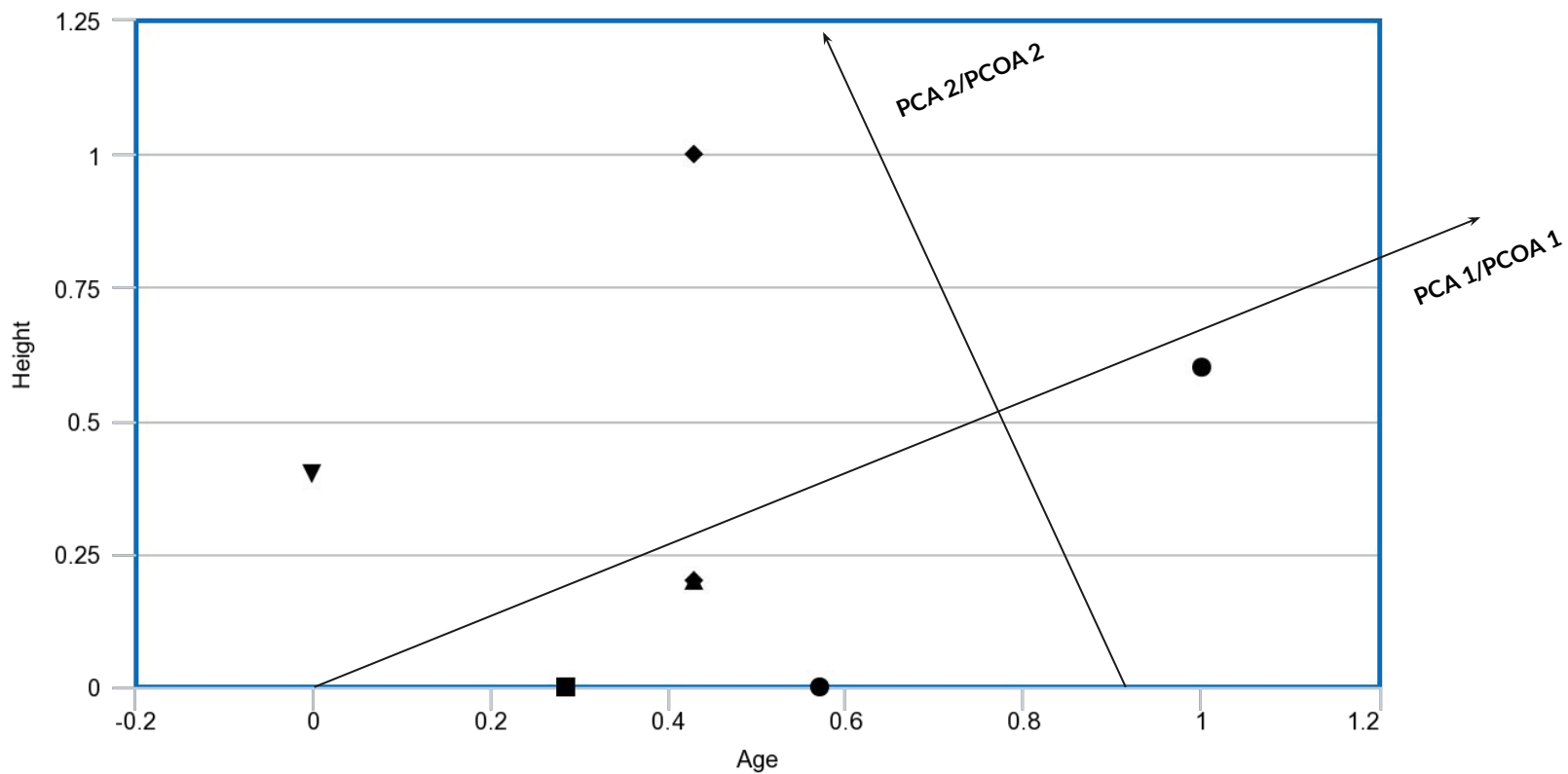
---

# Interpreting the plots

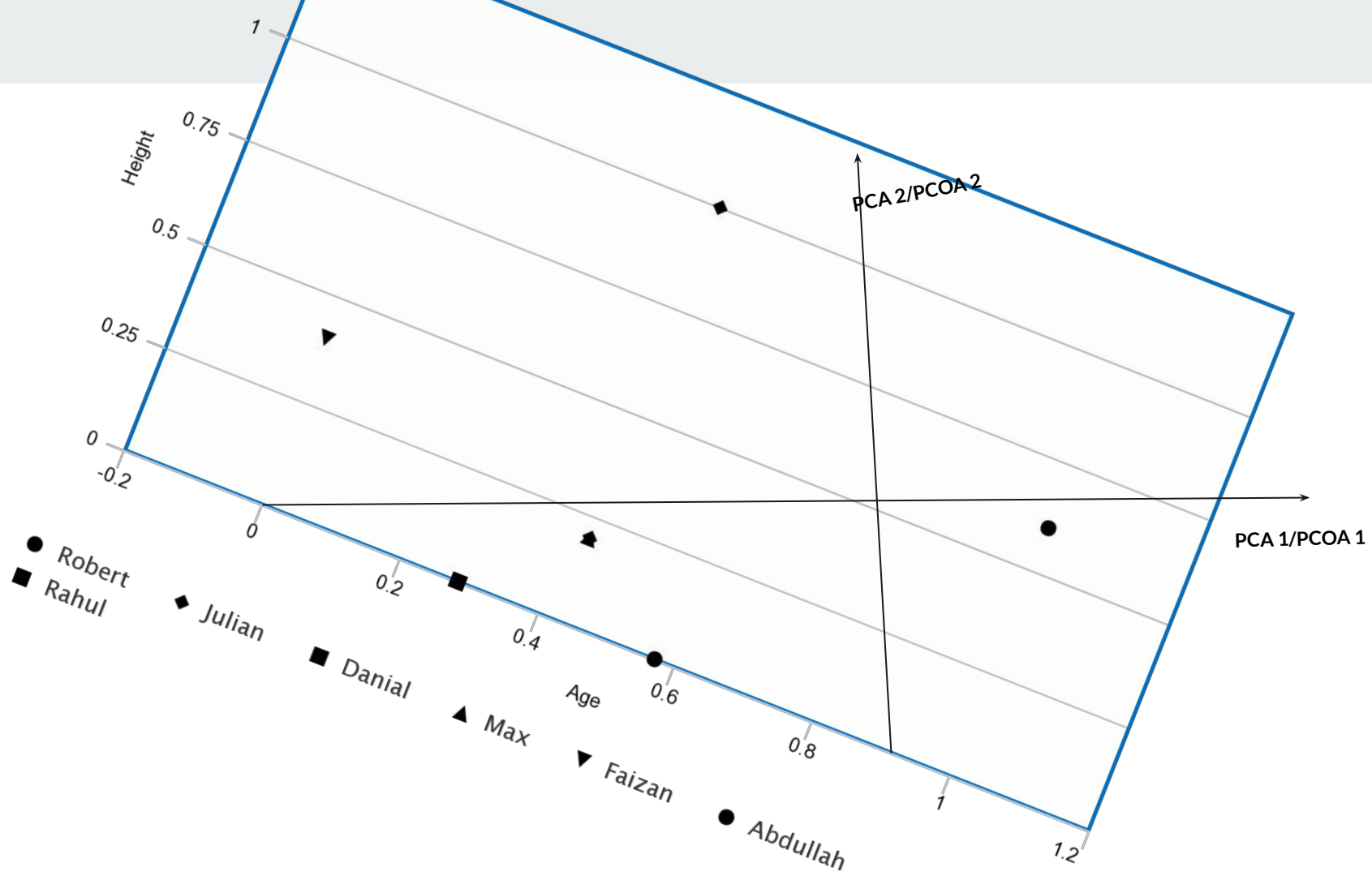


1. There is Principal Component/Coordinate for each dimensions
  - a. If we have “ $n$ ” variables, we would have “ $n$ ” Principal Components/Coordinates
2. PC1/PCoA1 would span the direction of most variation  
PC2/PCoA2 would span in the direction of 2<sup>nd</sup> most variation  
.  
.  
.  
PC“ $n$ ”/PCoA“ $n$ ” would span in the direction of “ $n$ ”<sup>th</sup> most variation
3. Each axis has an eigenvalue whose magnitude indicates the amount of variation captured in that axis

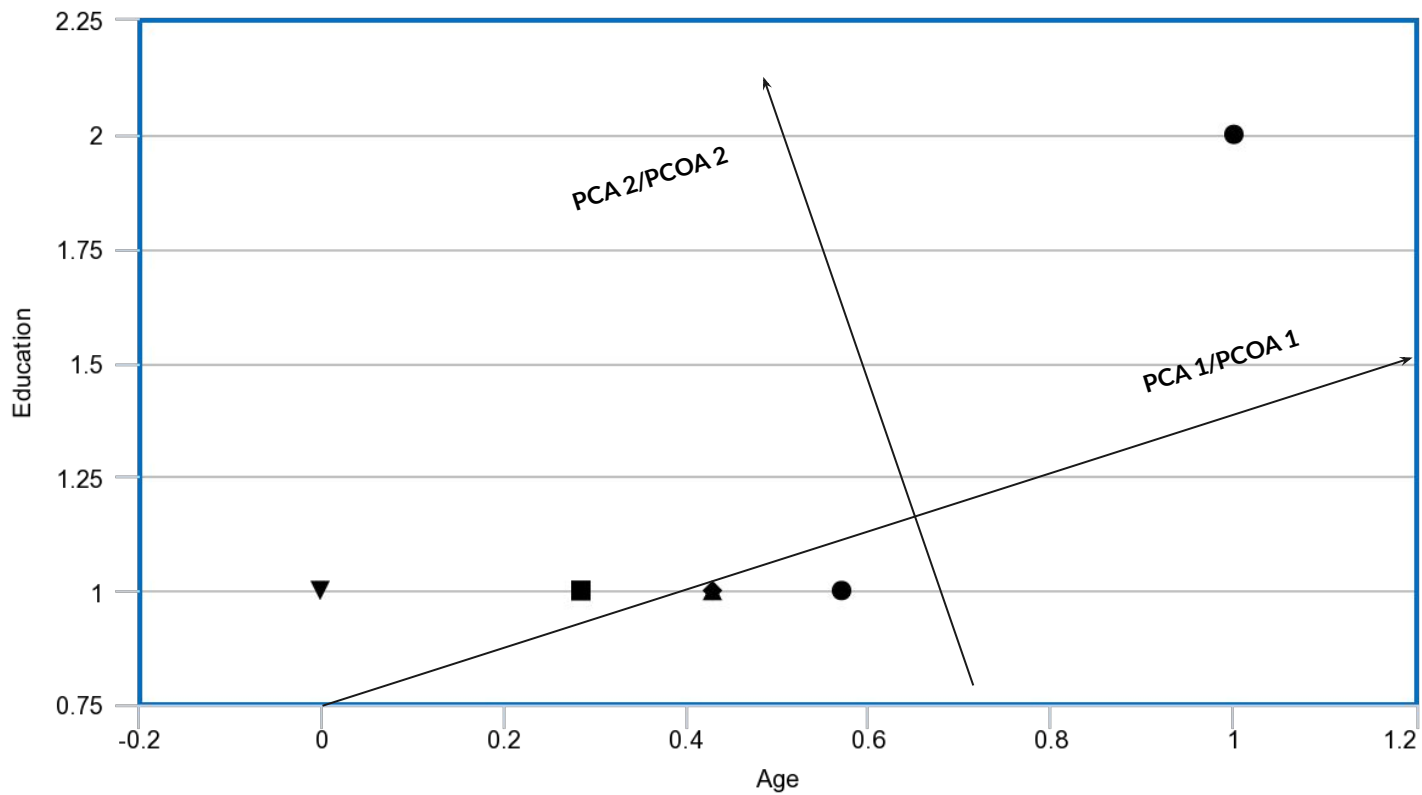
<u>Name(ID)</u>	<u>Age</u>		<u>Height</u>		<u>Gender</u> (1=f, 2=m, 3=other)	<u>Education Level</u> (0=Bachelor, 1= Master, 2= Post Doc)	<u>Class Label : Teacher(1) or Student(0)</u>
Robert	30	1	6.1	3/5	m(2)	Post Doc(2)	Teacher(1)
Julian	26	3/7	6.3	1	m(2)	Master(1)	Student(0)
Danial	25	2/7	5.8	0	m(2)	Master(1)	Student(0)
Max	26	3/7	5.9	1/5	m(2)	Master(1)	Student(0)
Faizan	23	0	6.0	2/5	m(2)	Master(1)	Student(0)
Abdullah	27	4/7	5.8	0	m(2)	Master(1)	Student(0)
Ammar	26	3/7	5.9	1/5	m(2)	Master(1)	Student(0)
Rahul	25	2/7	5.8	0	m(2)	Master(1)	Student(0)
<u>Mean</u>	26	3/7	5.95	0.3	2	1.125	



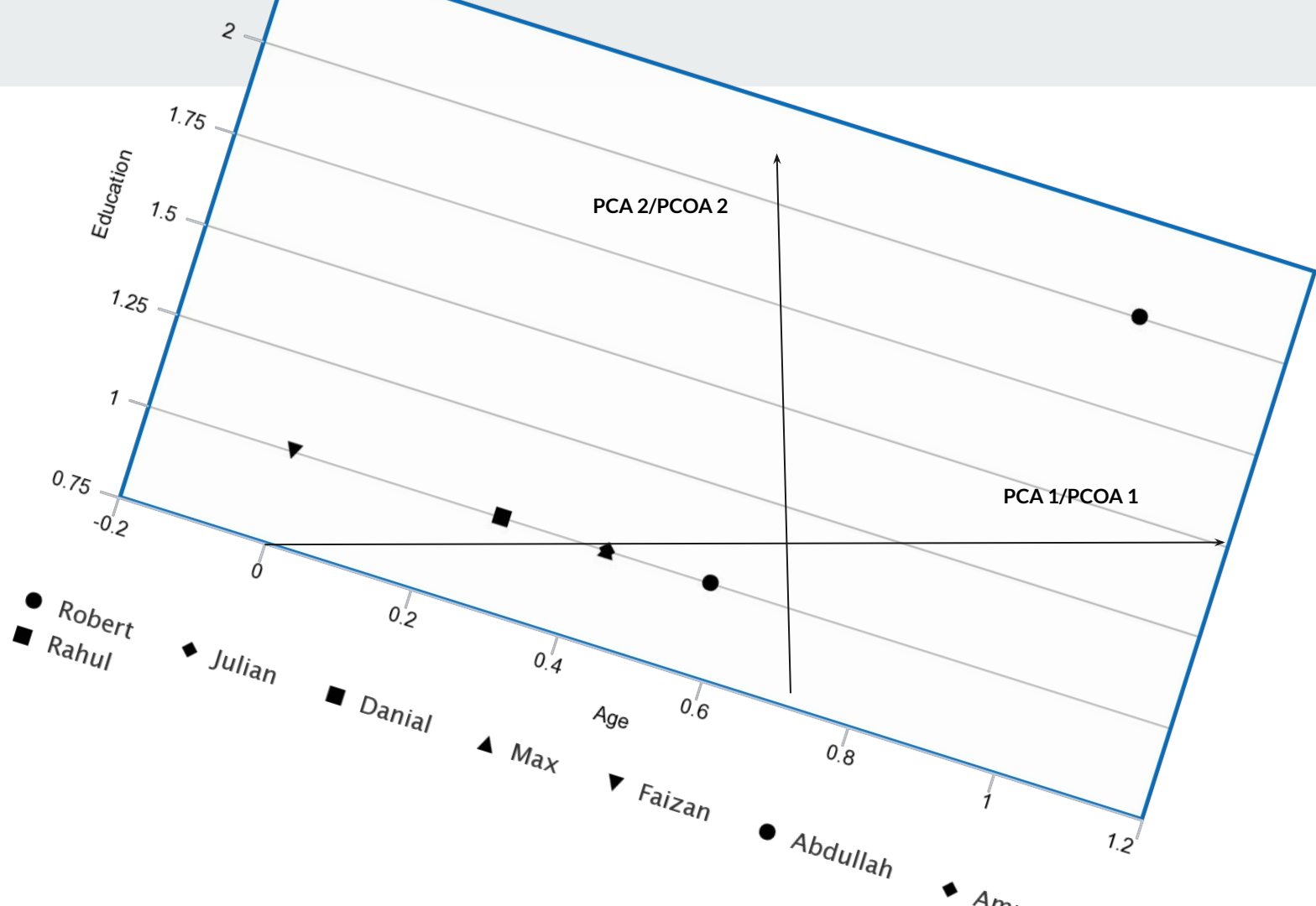
● Robert    ◆ Julian    ■ Danial    ▲ Max    ▼ Faizan    ● Abdullah    ◆ Ammar  
■ Rahul







● Robert    ◆ Julian    ■ Danial    ▲ Max    ▼ Faizan    ● Abdullah    ◆ Ammar  
■ Rahul



# Questions?

---

## Ordination Summary

Which ordination method should you choose?

If Euclidean distance and linear relationships are valid – PCA

e.g., most geological data types

Other distance measure more appropriate, but still linear – PCoA

e.g., biogeographic data

Other distance measure more appropriate; non-linear – NMDS

e.g., abundance count data (especially of species)

# NMDS (Non-metric Multidimensional Scaling)

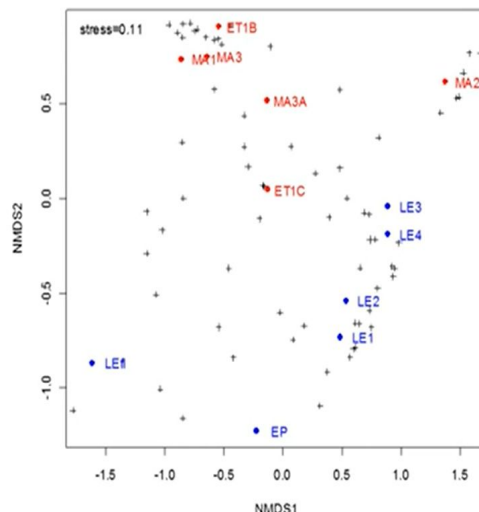
- Fundamentally different than PCA, CA (and DCA); more robust : **produces an ordination based on a distance or dissimilarity matrix.**
- Ordination based on **ranks** rather than **distance** rather than object A being 2.1 units distant from object B and 4.4 units distant from object C, object C is the "first" most distant from object A while object C is the "second" most distant.
- Avoids assumption of linear relationships among variables

## Placing Objects Initially

- Random Placement
- Placement according to a PCA result**
- Placement according to geographic distances
- Placement by moving from high to low dimensionality

## Interpreting NMDS Plots

Like other ordination plots, you should qualitatively identify gradients corresponding to underlying processes



### Differences from eigenanalysis:

- Does not extract components (based only on distance) so axes are meaningless\*
- Plot can be rotated, translated, or scaled as long as relative distances are maintained

\*metaMDS in vegan performs PCA rotation on the results so that axis 1 contains the greatest variance

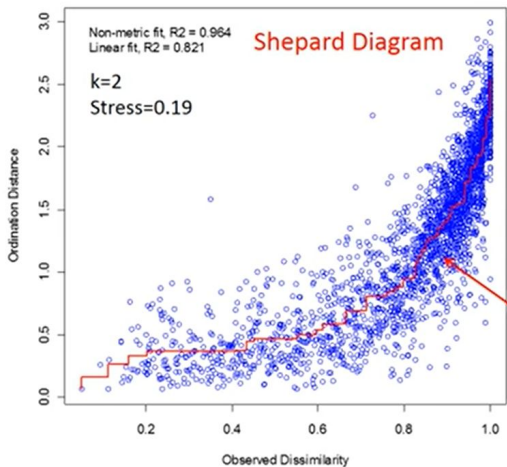
# NMDS (Non-metric Multidimensional Scaling)

## Stress

NMDS Maximizes rank-order correlation between distance measures and distance in ordination space. Points are iteratively moved to **minimize "stress"**. Stress is a measure of the mismatch between the two kinds of distance.

### NMDS Goodness-of-Fit

Goodness-of-fit is measured by “stress” – a measure of rank-order disagreement between observed and fitted distances



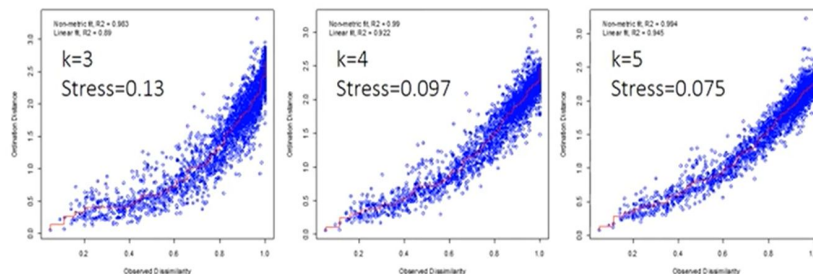
Stress calculated from residuals around monotone regression line

Ideally, all points should fall on monotonic line (increasing ordination distance = increasing observed distance)

Think of optimizing stress as: “Pulling on all points a little bit so no single point is completely wrong, all points are a little off compared to distances”

### NMDS Goodness-of-Fit

Stress *always* decreases with increasing dimensionality  $k$



Remember that a 2D solution is not a projection of higher-dimensional solutions (as in PCA)

## Shepard Diagram

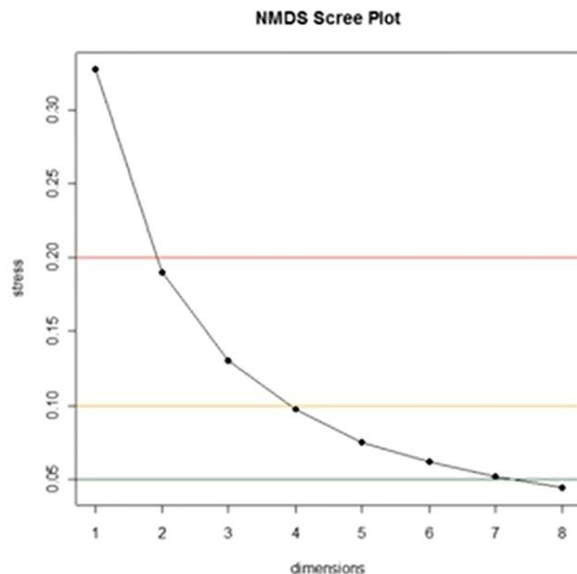
# NMDS (Non-metric Multidimensional Scaling)

## NMDS Goodness-of-Fit

As in PCA, can construct a scree plot of stress vs. dimensionality

In practice, people normally do ordination in 2 or 3 dimensions

### Scree Plot



Goodness of fit:

>0.2 Poor (risks in interpretation)

0.1-0.2 Fair (some distances misleading)

0.05-0.1 Good (inferences confident)

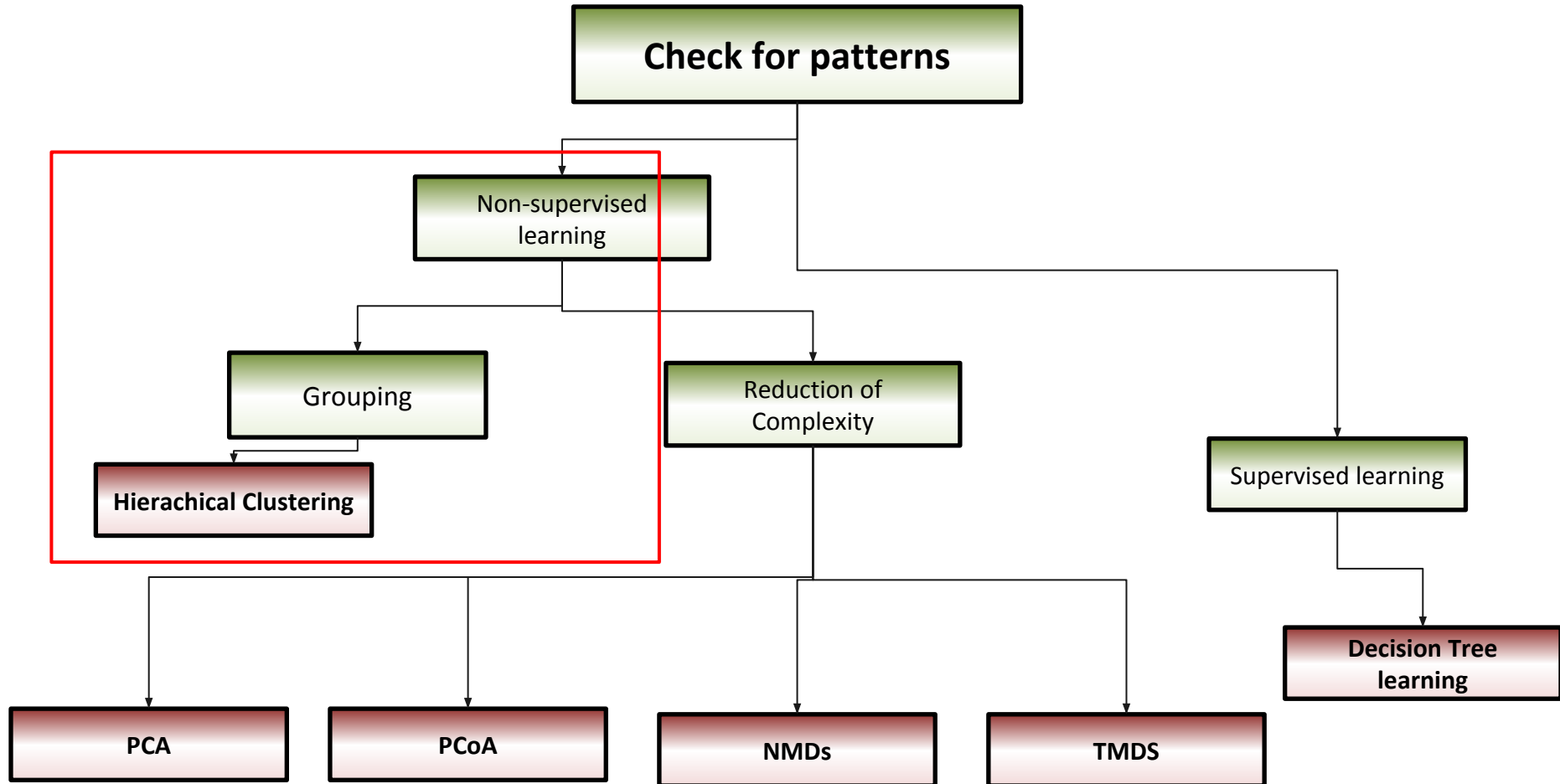
<0.05 Excellent

---

# Grouping

- Clustering
  - Centroid Based
    - K-Means
  - Density Based
    - DBSCAN
  - Hierarchical
    - Agglomerative

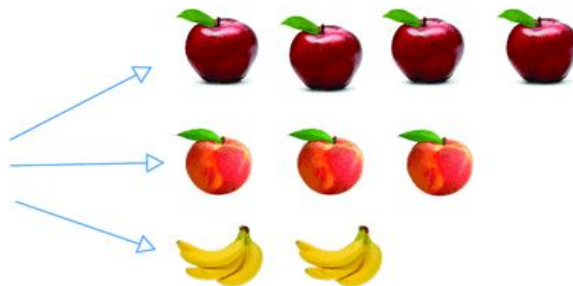




# Clustering

unsupervised learning

Input data



finding a **structure** in a collection of **unlabeled data** i.e. the process of **organizing objects into groups** whose members are similar in some way

why?

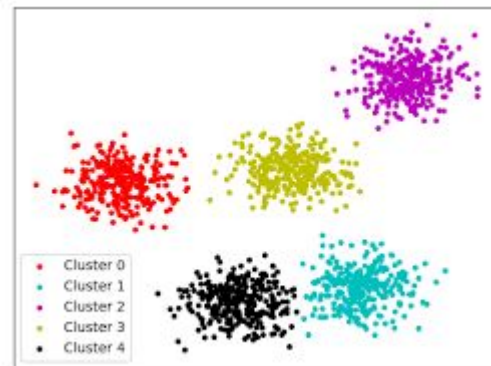
finding representatives for

- homogeneous groups (**data reduction**),
- in finding “natural clusters” and describe their unknown properties (**“natural” data types**),
- in finding useful and suitable groupings (**“useful” data classes**) or
- in finding unusual data objects (**outlier detection**)

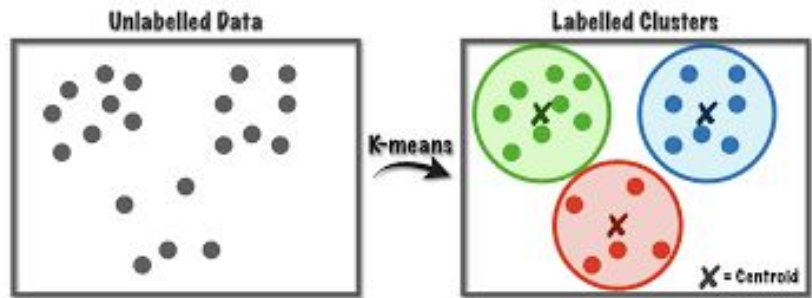
how?

- Centroid based :  
**K-Means**
- Density based :  
**DBSCAN**
- Hierarchical :  
**Agglomerative**

what?



# Centroid Based Clustering



**K-Means**

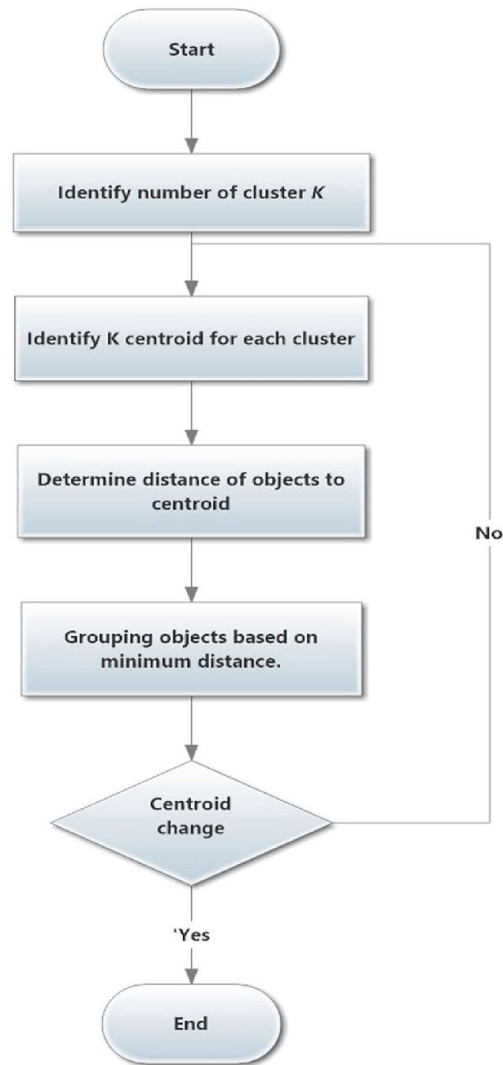
## Centroid

The middle of a cluster i.e. a multidimensional average of a cluster

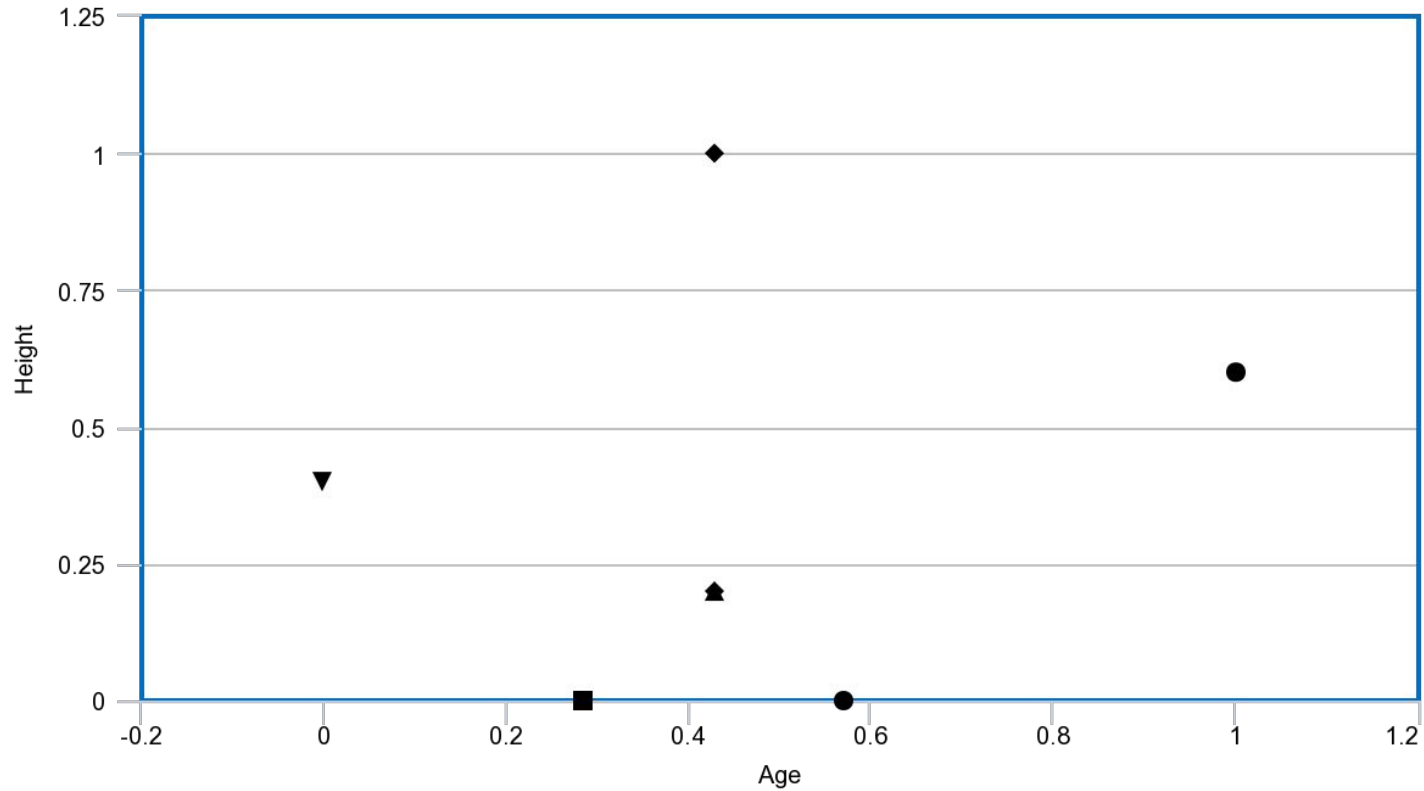
## why & why not?

- + simple
- + guarantees convergence
- + scales to large data set
- clustering goodness depends on initialization
- sensitive to outliers
- troubled with clusters of varying size & density

how?

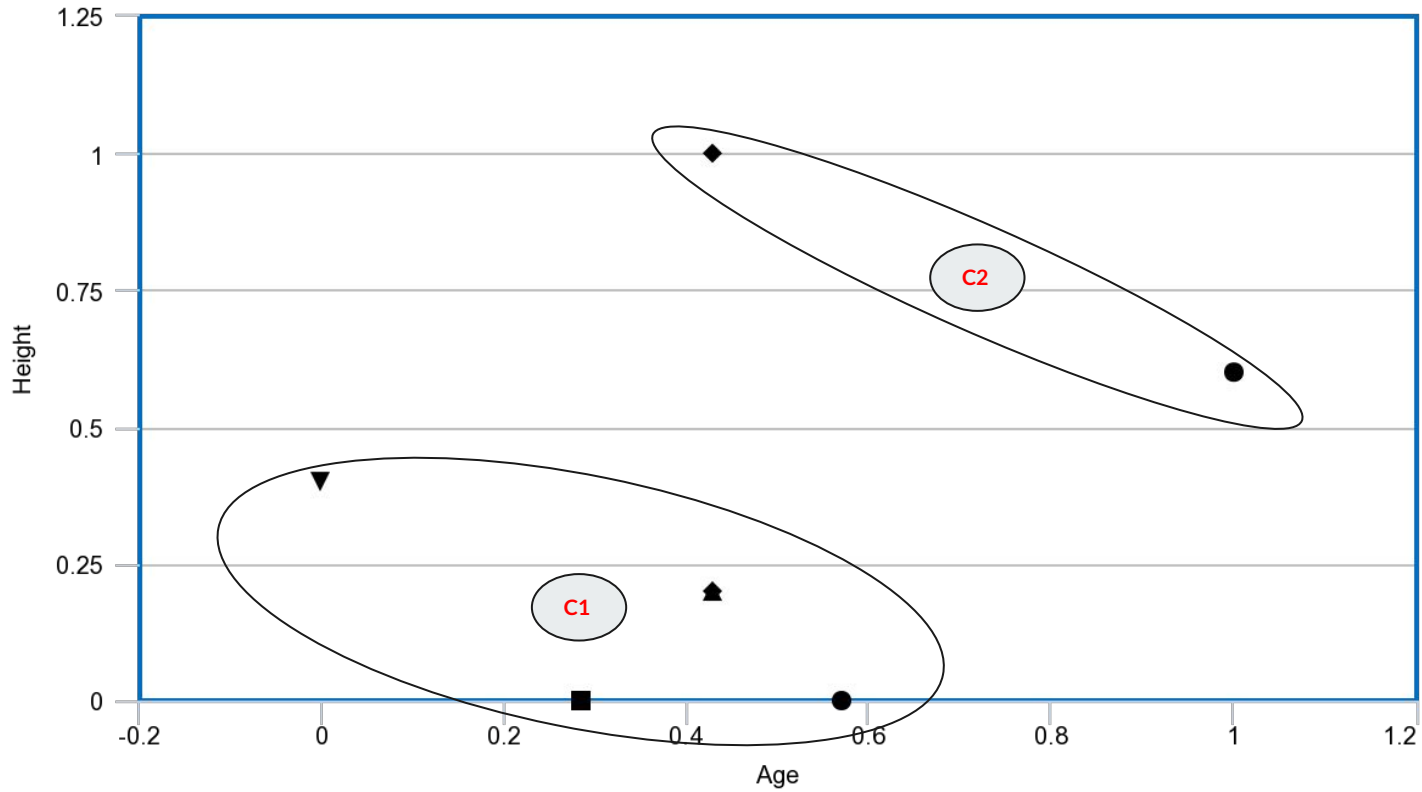


# Centroid Based Clustering: K-Means



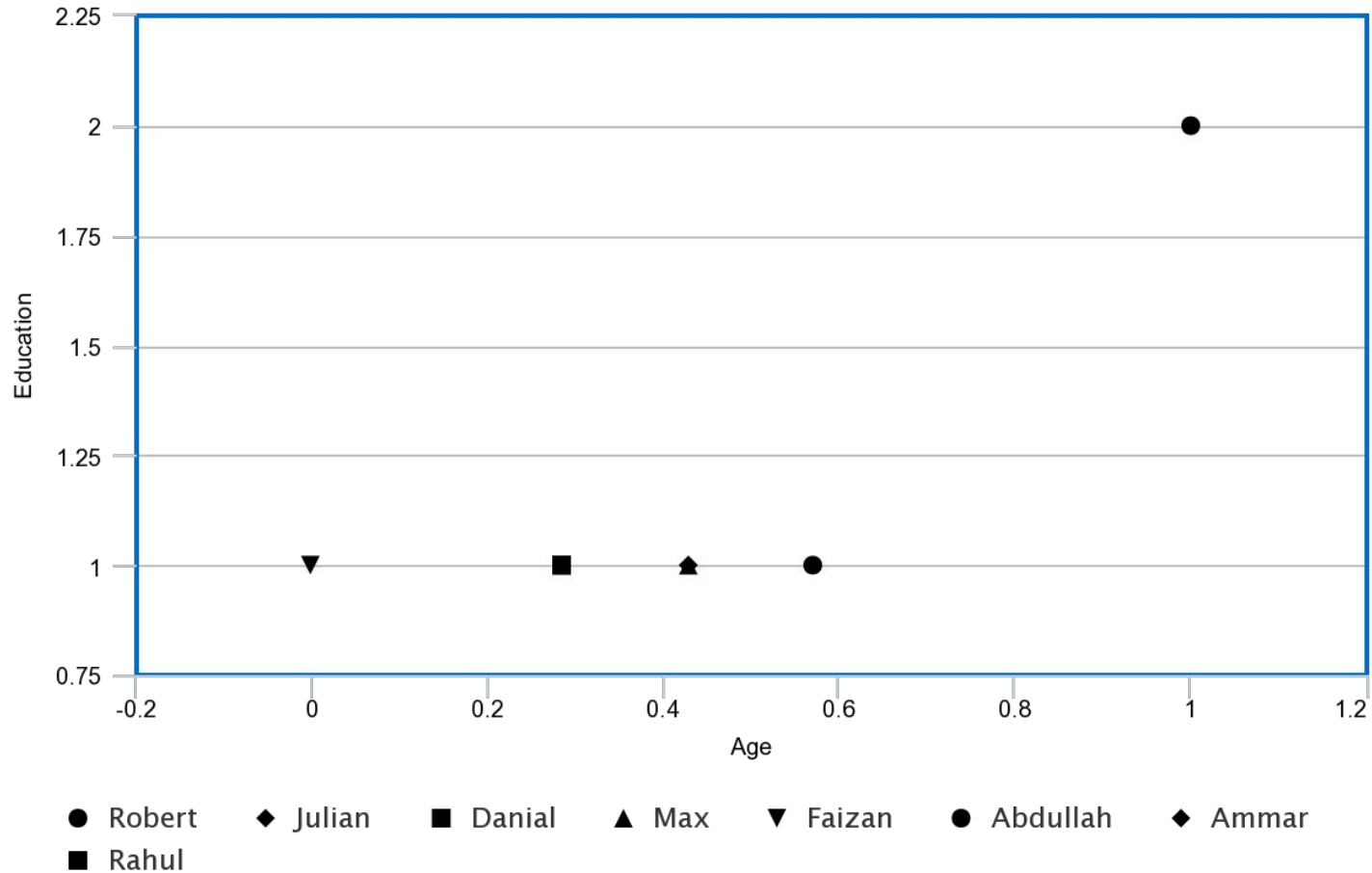
● Robert    ◆ Julian    ■ Danial    ▲ Max    ▼ Faizan    ● Abdullah    ◆ Ammar  
■ Rahul

# Centroid Based Clustering: K-Means

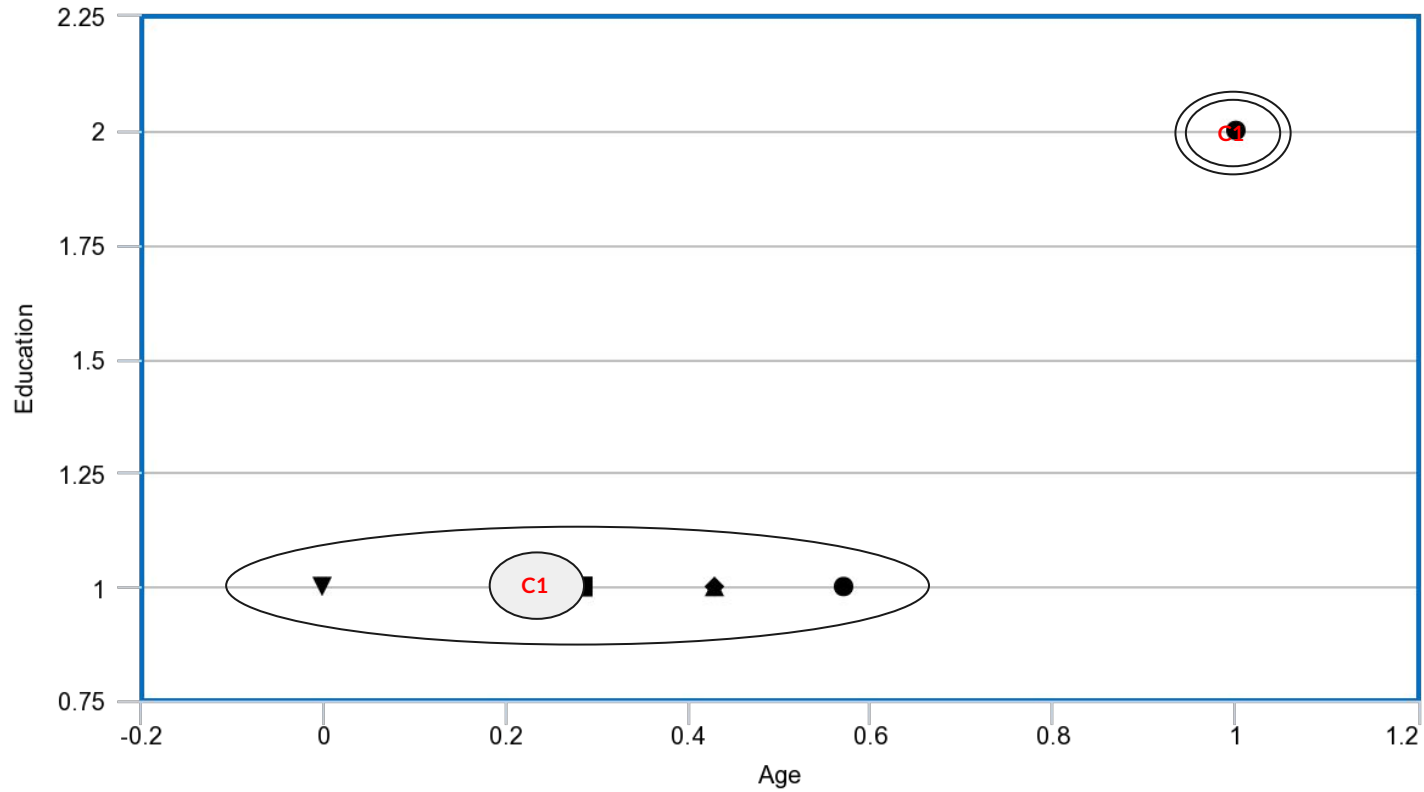


● Robert    ◆ Julian    ■ Danial    ▲ Max    ▼ Faizan    ● Abdullah    ◆ Ammar  
■ Rahul

# Centroid Based Clustering: K-Means



# Centroid Based Clustering: K-Means



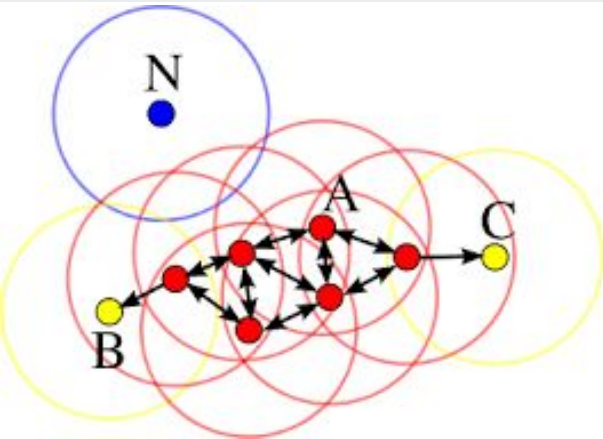
● Robert    ◆ Julian    ■ Danial    ▲ Max    ▼ Faizan    ● Abdullah    ◆ Ammar  
■ Rahul

# Questions?

---



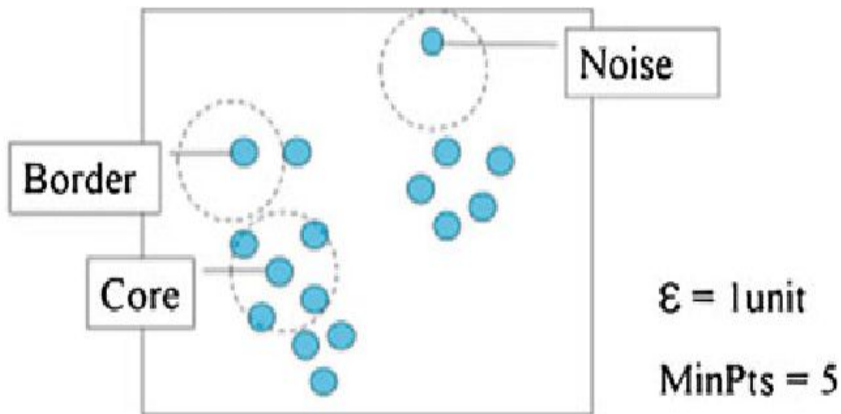
# Density Based Clustering: DBSCAN



how?

- Arbitrary select a point  $p$
- Retrieve all points density-reachable from  $p$  w.r.t.  $Eps$  and  $MinPts$ .
- If  $p$  is a core point, a cluster is formed.
- If  $p$  is a border point, no points are density-reachable from  $p$  and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

core, border & noise points

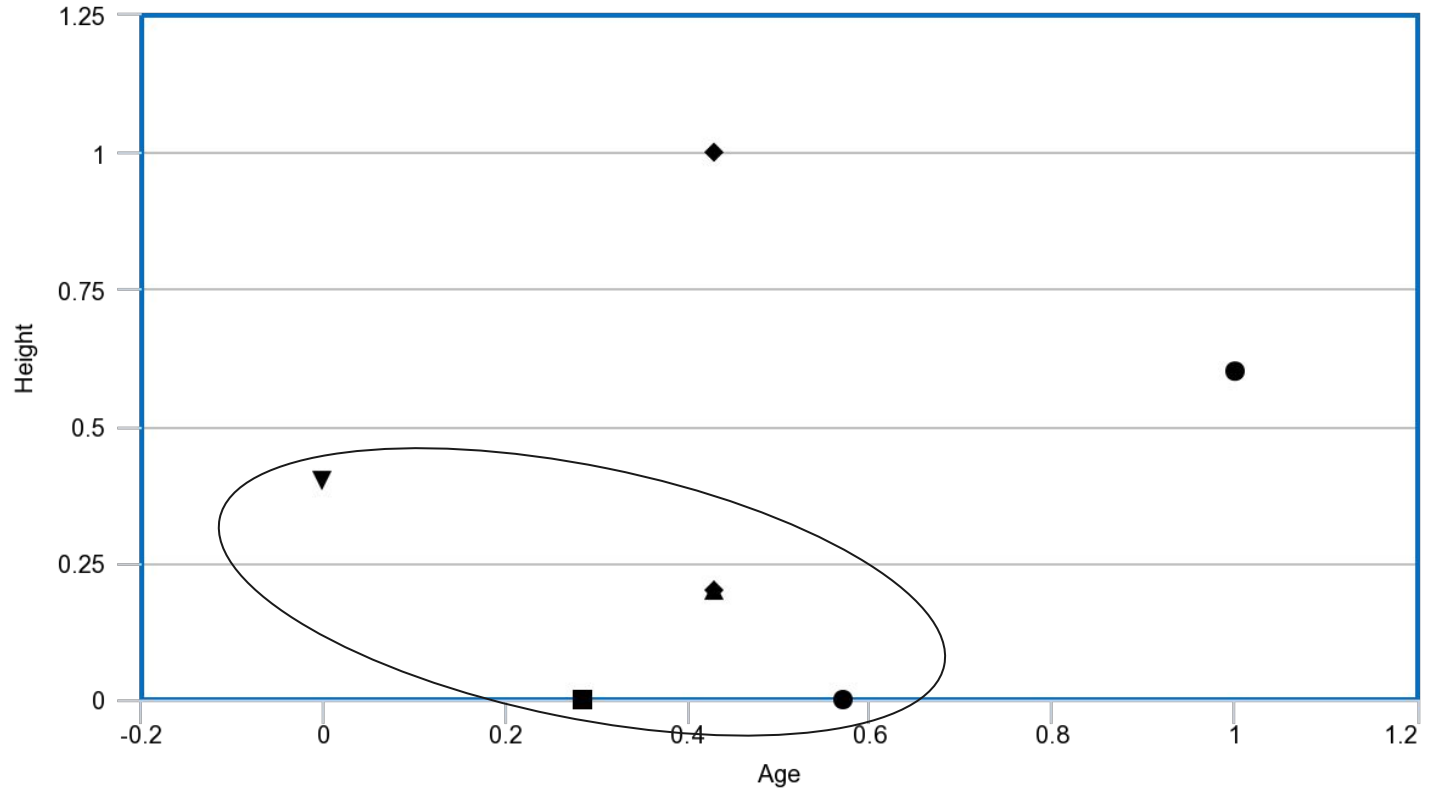


why & why not?

- + can handle varying density
- + good with outliers
- does not work well at similar densities
- struggles with high dimensional data

# Density Based Clustering: DBSCAN

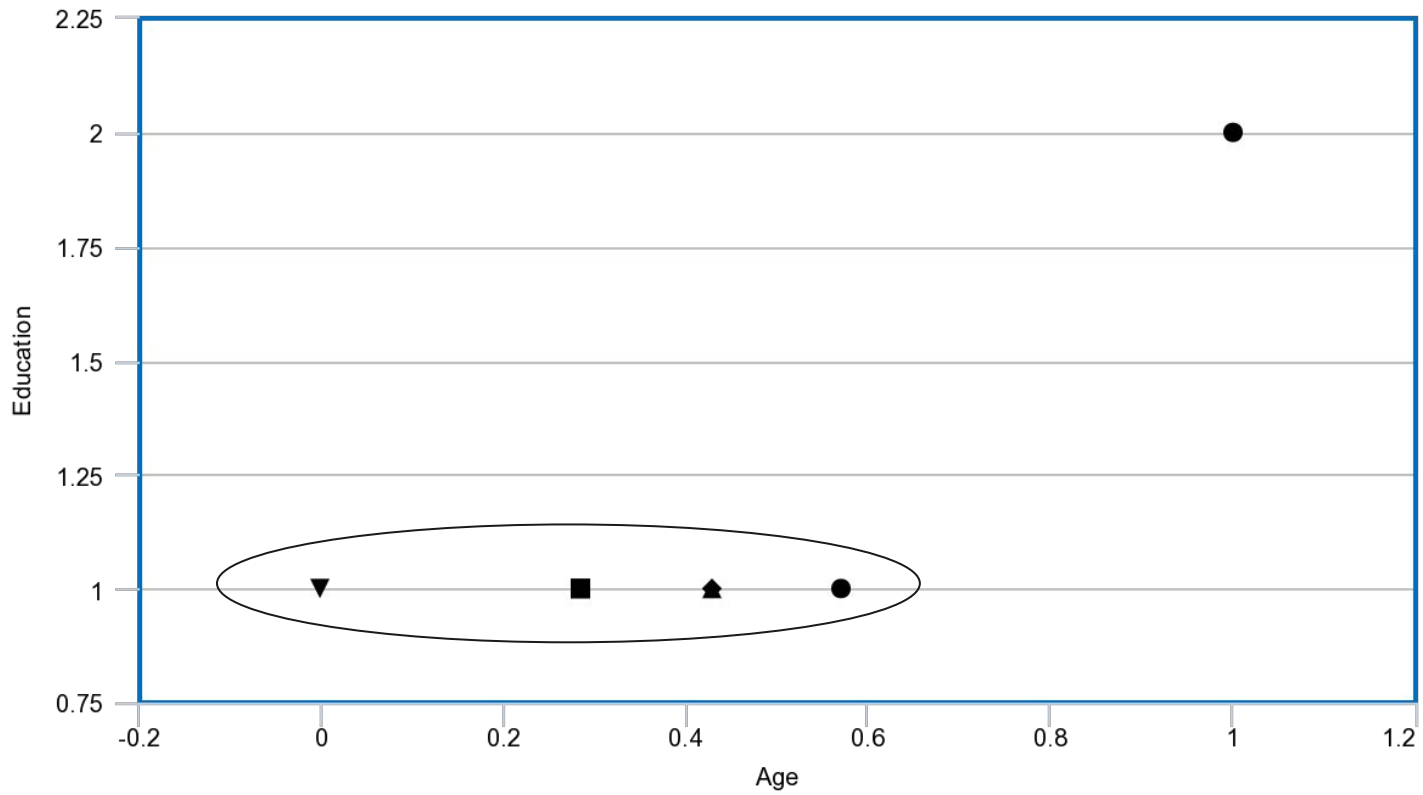
epsilon: 0.5  
minPts: 2



● Robert    ◆ Julian    ■ Danial    ▲ Max    ▼ Faizan    ● Abdullah    ◆ Ammar  
■ Rahul

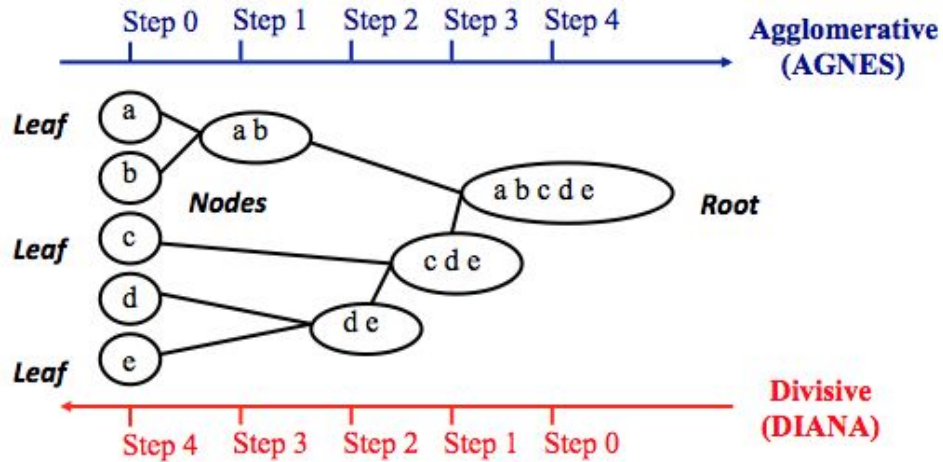
# Density Based Clustering: DBSCAN

epsilon: 0.5  
minPts: 2

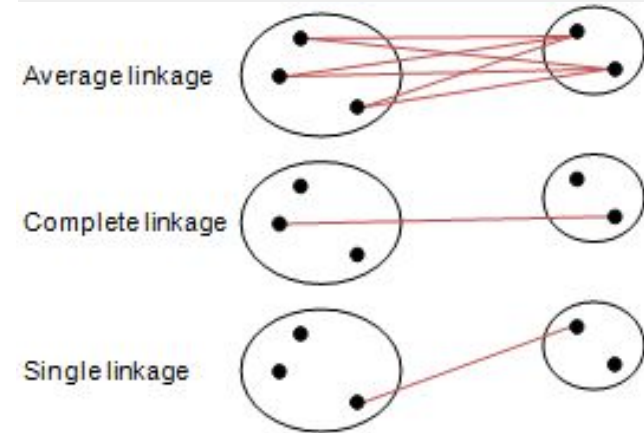


- Robert
- ◆ Julian
- Danial
- ▲ Max
- ▼ Faizan
- Abdullah
- ◆ Ammar
- Rahul

# Hierarchical Clustering: Agglomerative



## linkage criterion



## why & why not?

- + easy to do & understand
- + possibilities to choose from a hierarchy of clusters
- possible to misinterpret
- expensive

Hierarchical cluster analysis is an algorithmic approach to find discrete groups with varying degrees of (dis)similarity in a data set represented by a (dis)similarity matrix. These groups are hierarchically organised as the algorithms proceed and may be presented as a **dendrogram**

# Hierarchical Clustering: Dendrogram

