LECTURE 11

AGENDA FOR TODAY

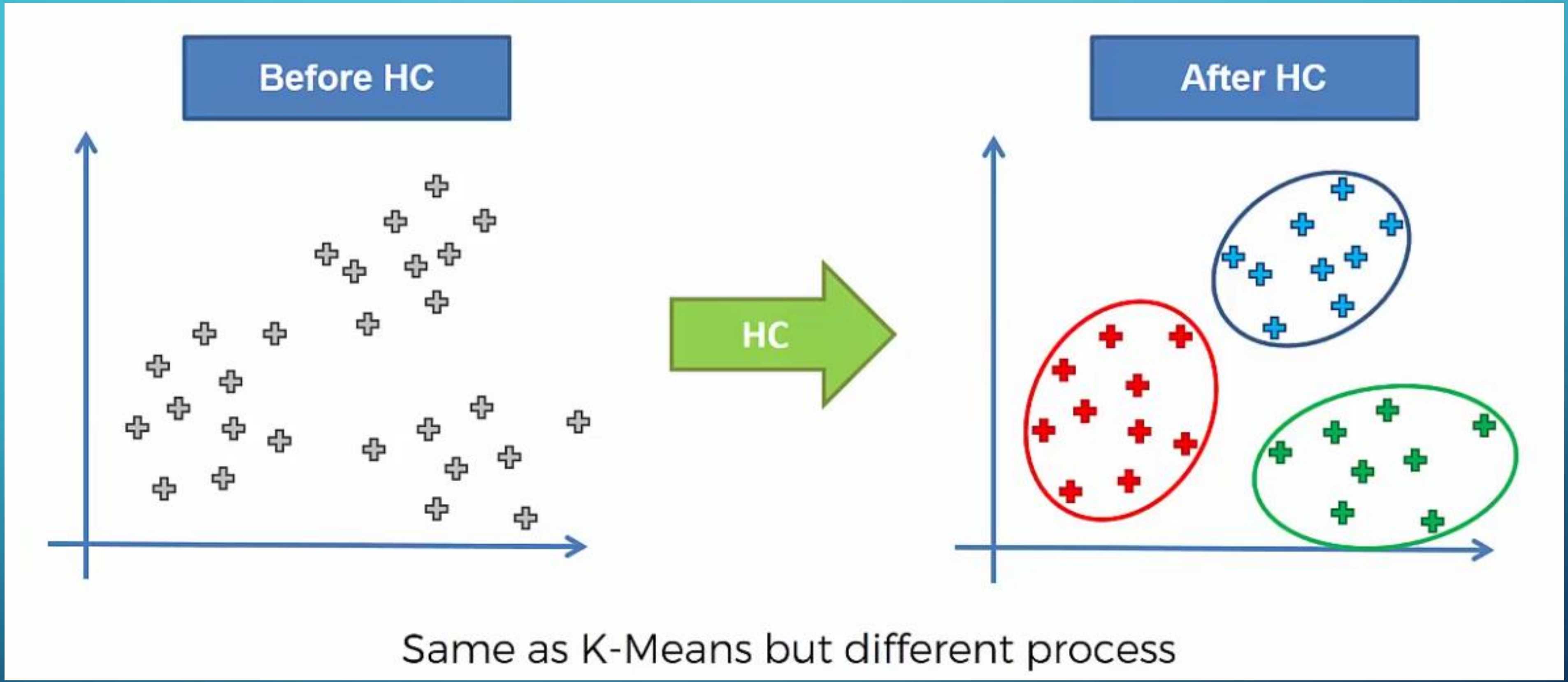
- Hierarchical Clustering Algorithm
- Principle Component Analysis
- Apriori

HIERARCHICAL CLUSTERING

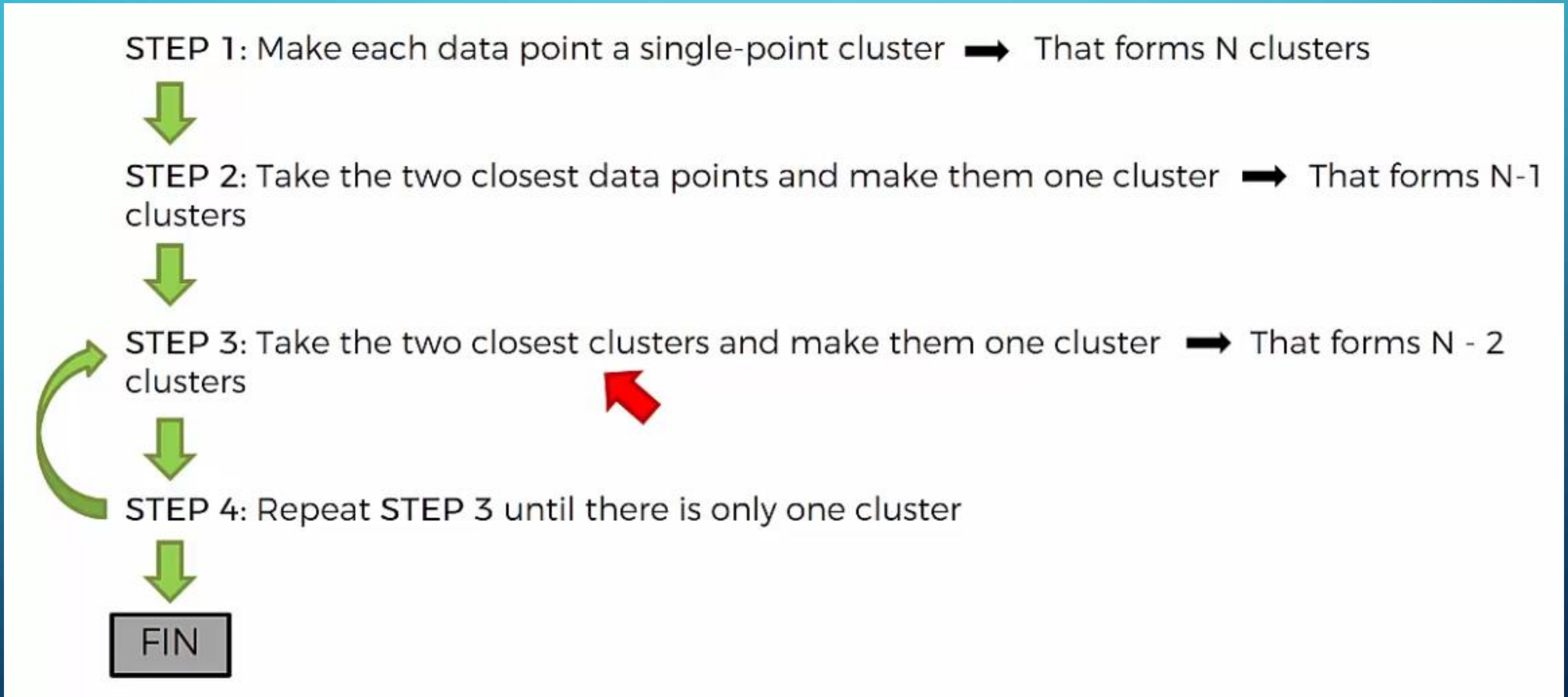
IT'S A UNSUPERVISED LEARNING ALGORITH

Recall K-Means Algorithm!!!

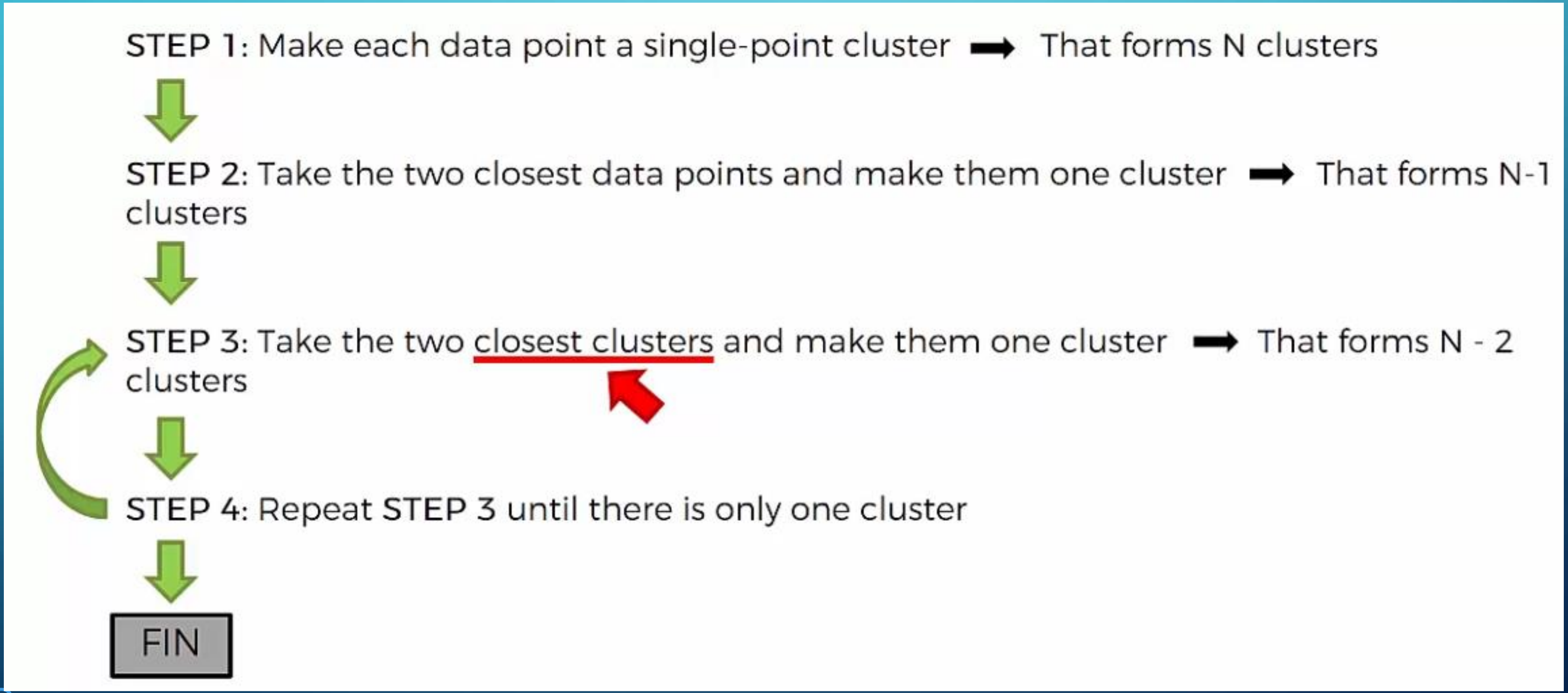
WHAT IT DOES FOR YOU?


Before HC
HC
After HC
Same as K-Means but different process

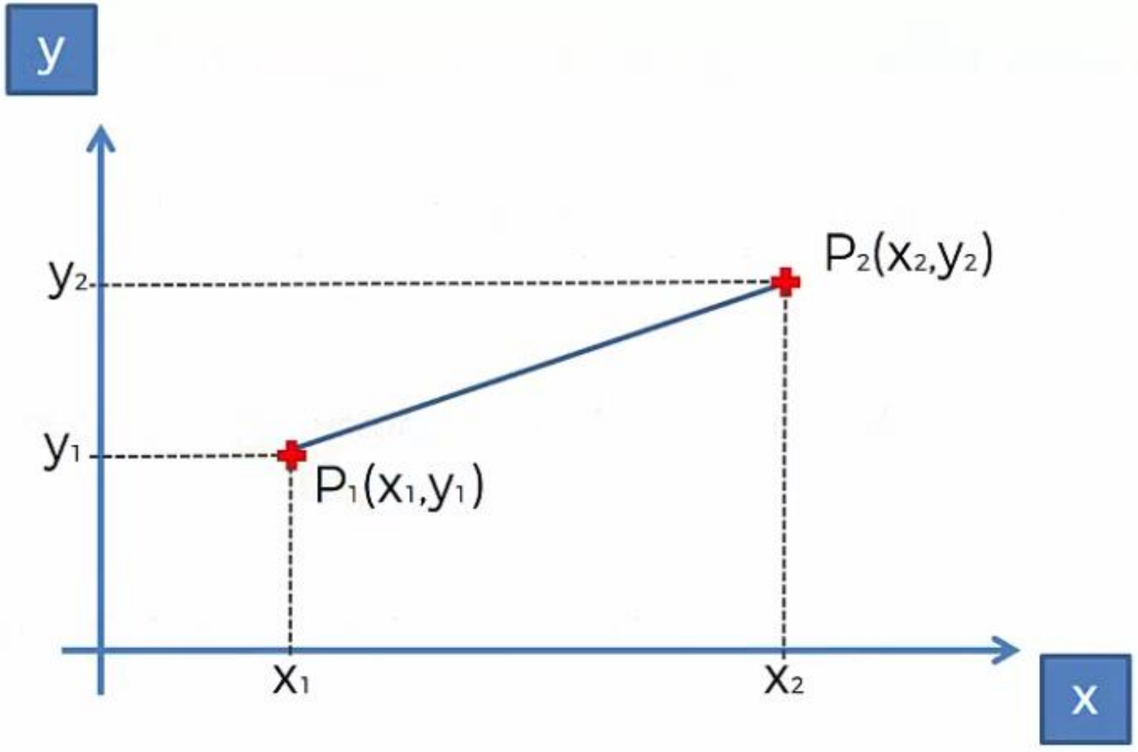
STEPS:


STEP 1: Make each data point a single-point cluster ➡ That forms N clusters
STEP 2: Take the two closest data points and make them one cluster ➡ That forms N-1 clusters
STEP 3: Take the two closest clusters and make them one cluster ➡ That forms N - 2 clusters
STEP 4: Repeat STEP 3 until there is only one cluster
FIN

HOW TO FIND CLOSEST CLUSTERS?

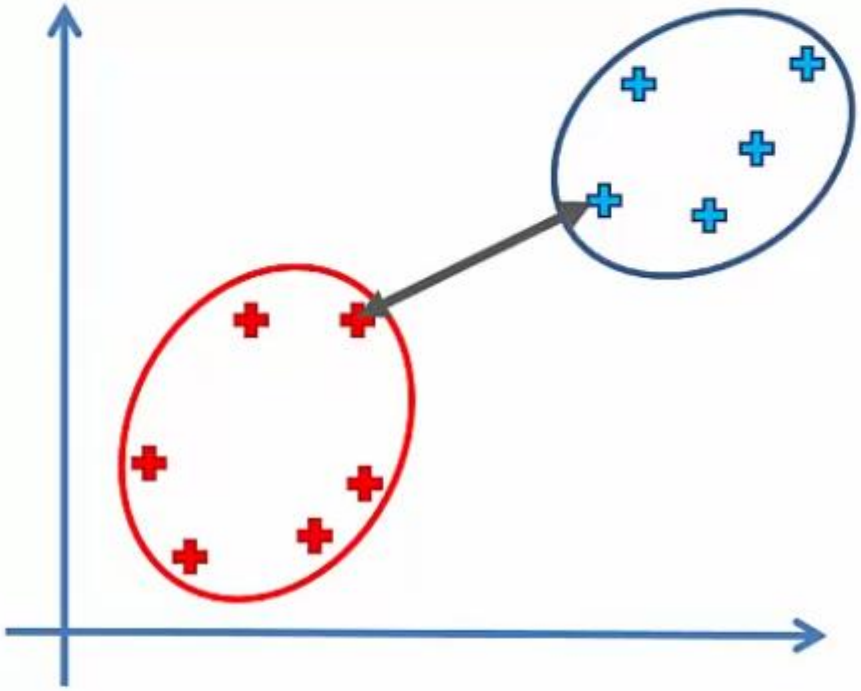


EUCLIDEAN DISTANCE TO THE RESCUE!!

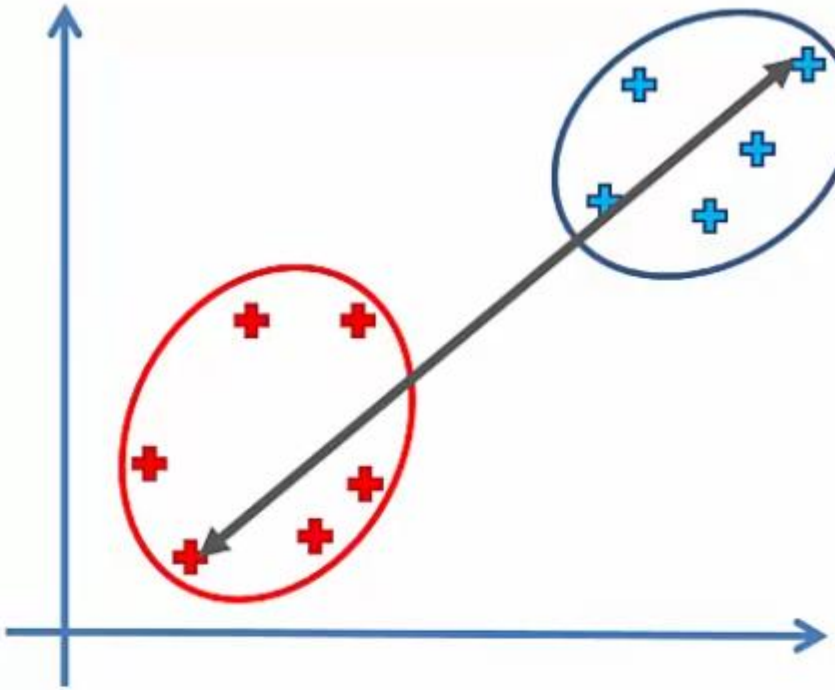


Euclidean Distance between P_1 and $P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Distance Between Two Clusters:

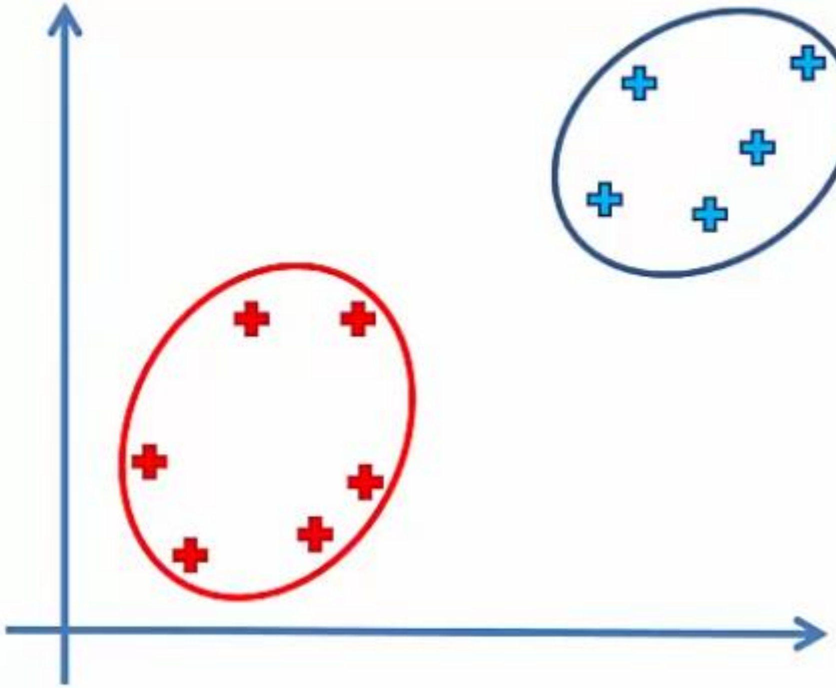


- Option 1: Closest Points



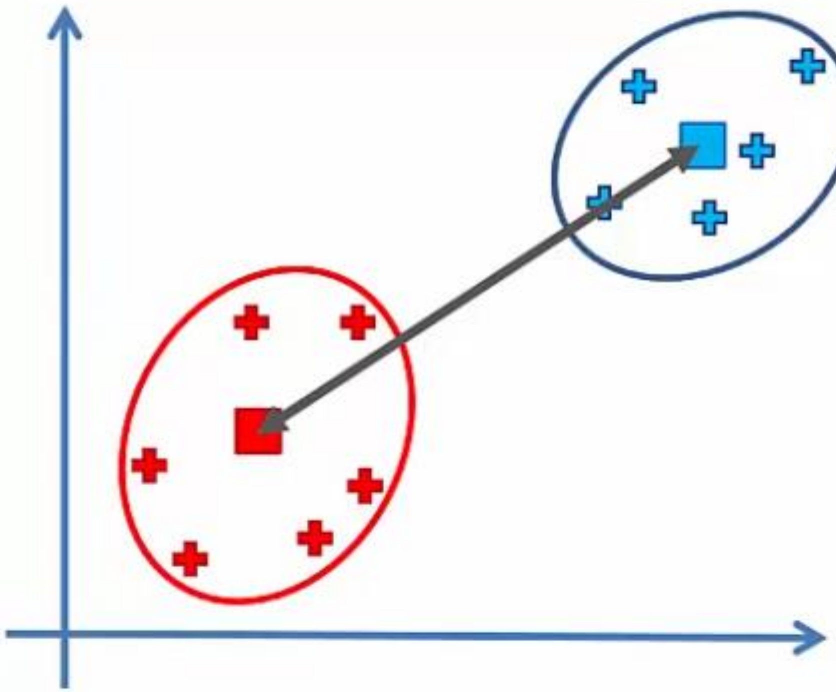
Distance Between Two Clusters:

- Option 1: Closest Points
- Option 2: Furthest Points



Distance Between Two Clusters:

- Option 1: Closest Points
- Option 2: Furthest Points
- Option 3: Average Distance

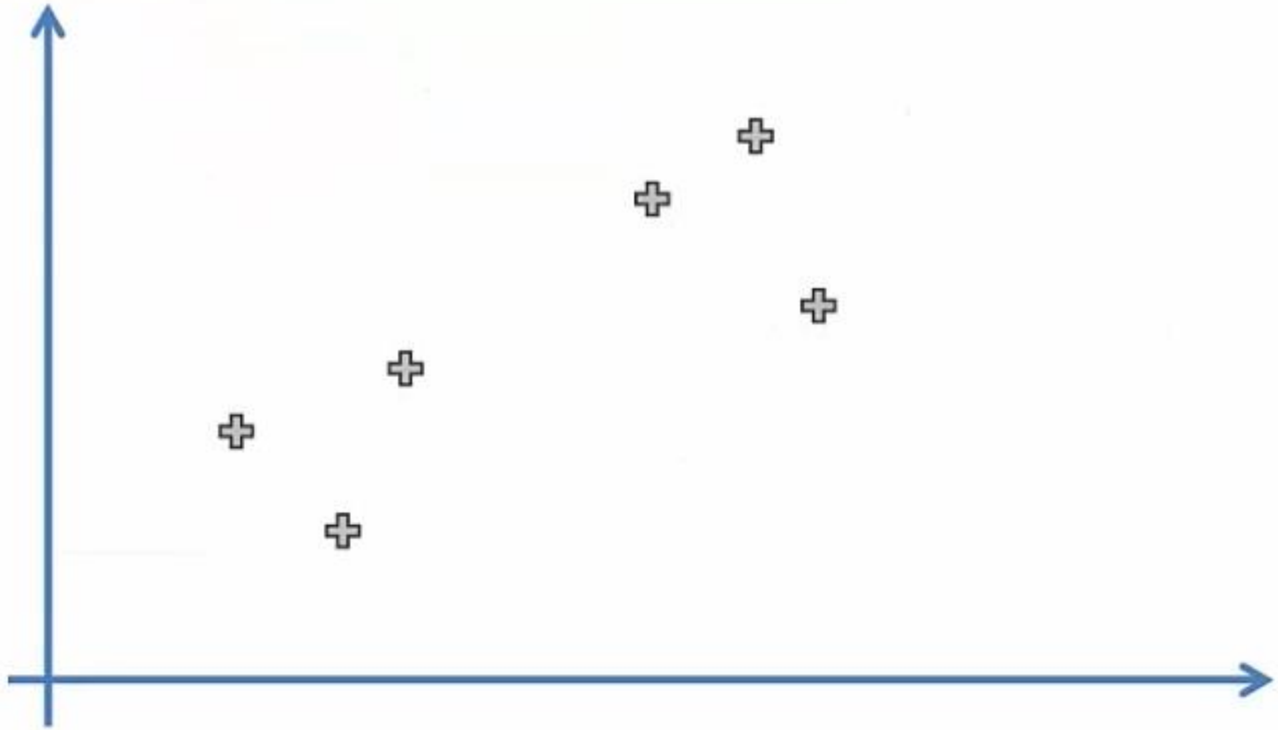


Distance Between Two Clusters:

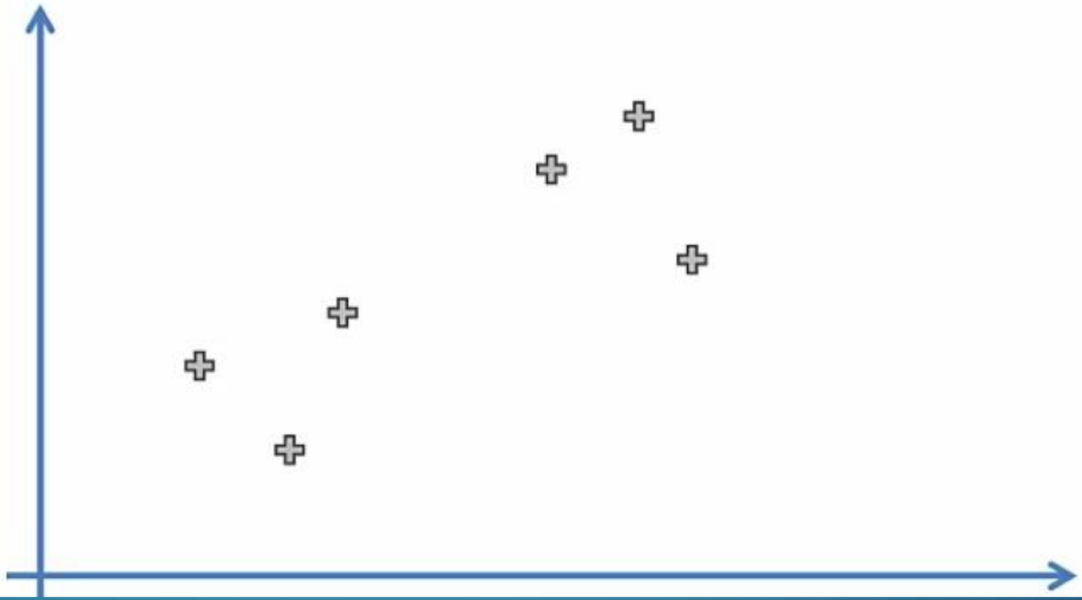
- Option 1: Closest Points
- Option 2: Furthest Points
- Option 3: Average Distance
- Option 4: Distance Between Centroids

HOW ALGORITHM WORKS??

Consider the following dataset of N = 6 data points

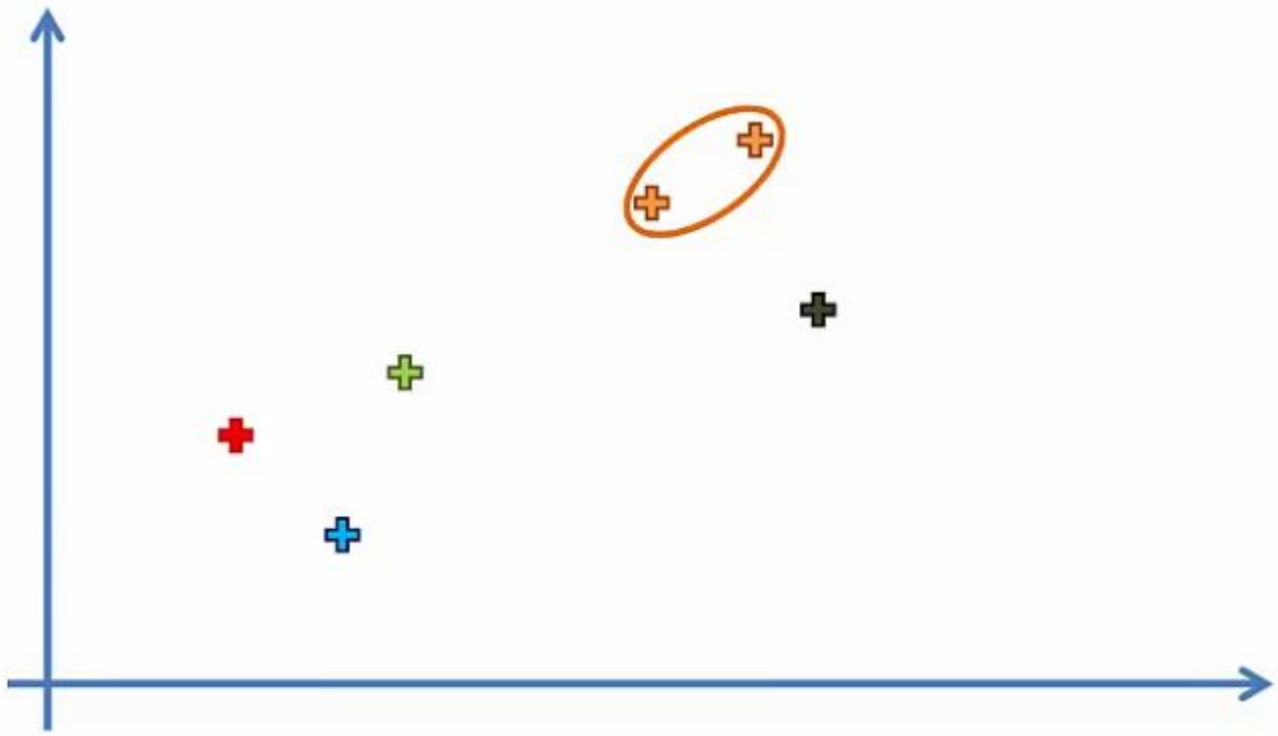


STEP 1: Make each data point a single-point cluster ➡ That forms 6 clusters



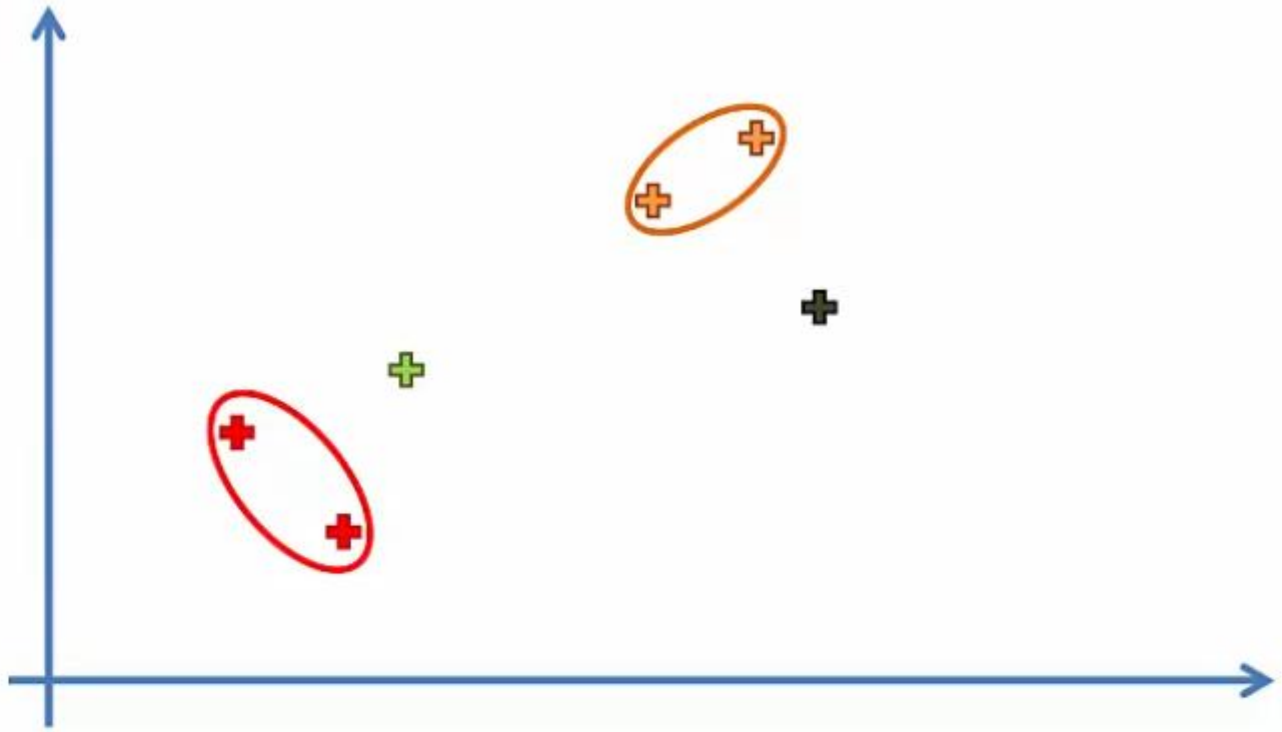
STEP 2: Take the two closest data points and make them one cluster

➡ That forms 5 clusters

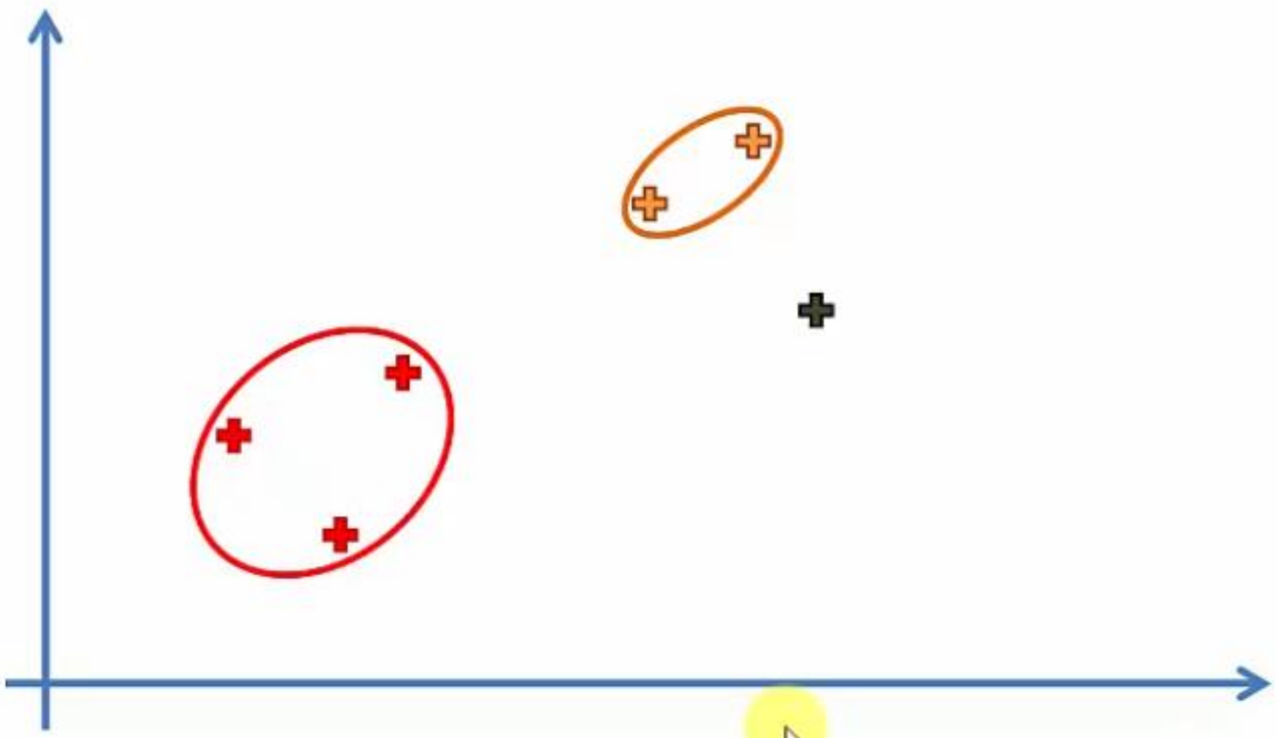


STEP 3: Take the two closest clusters and make them one cluster

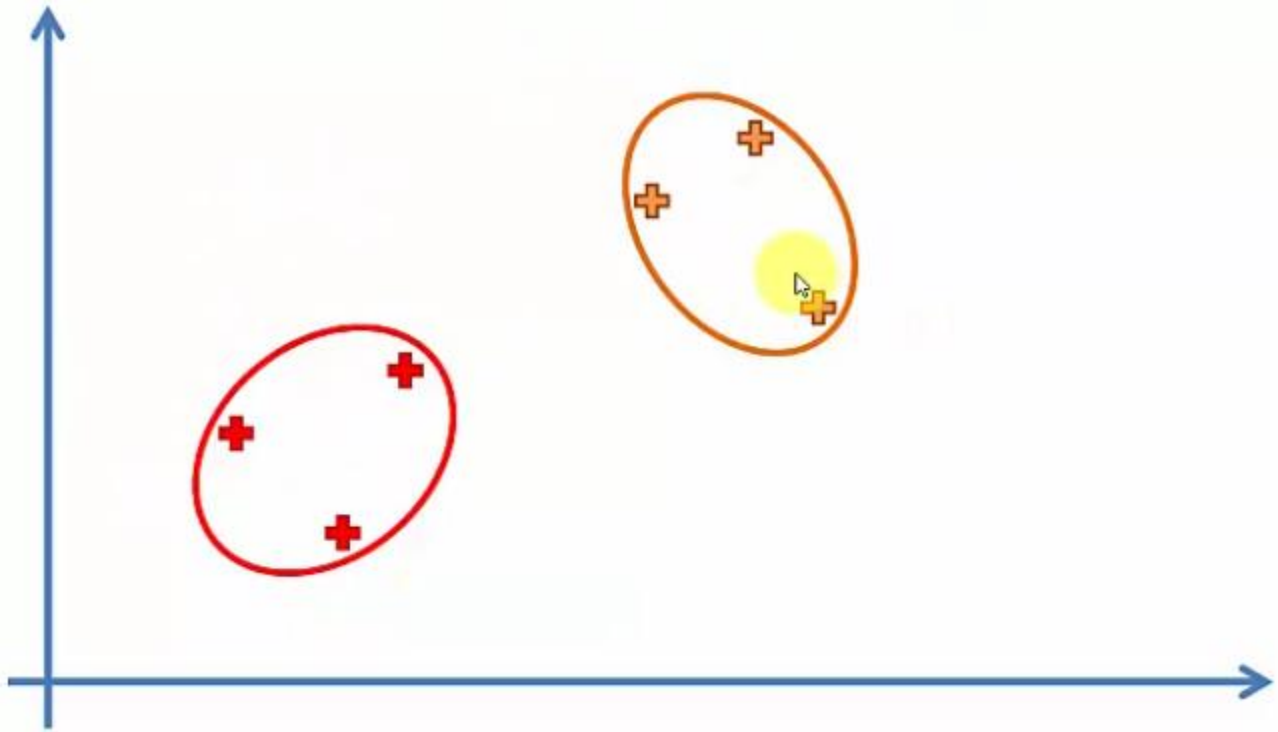
➡ That forms 4 clusters

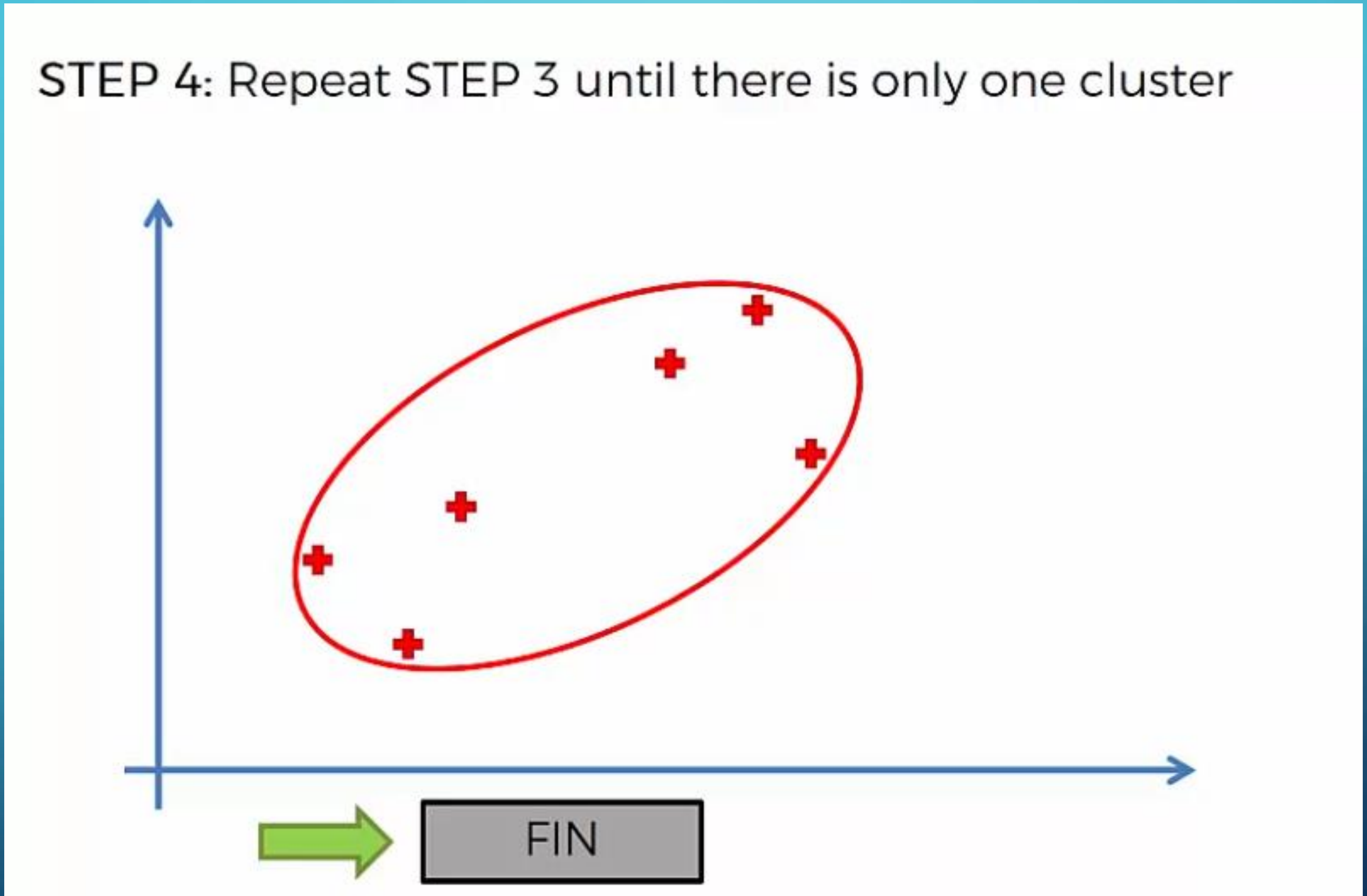


STEP 4: Repeat STEP 3 until there is only one cluster



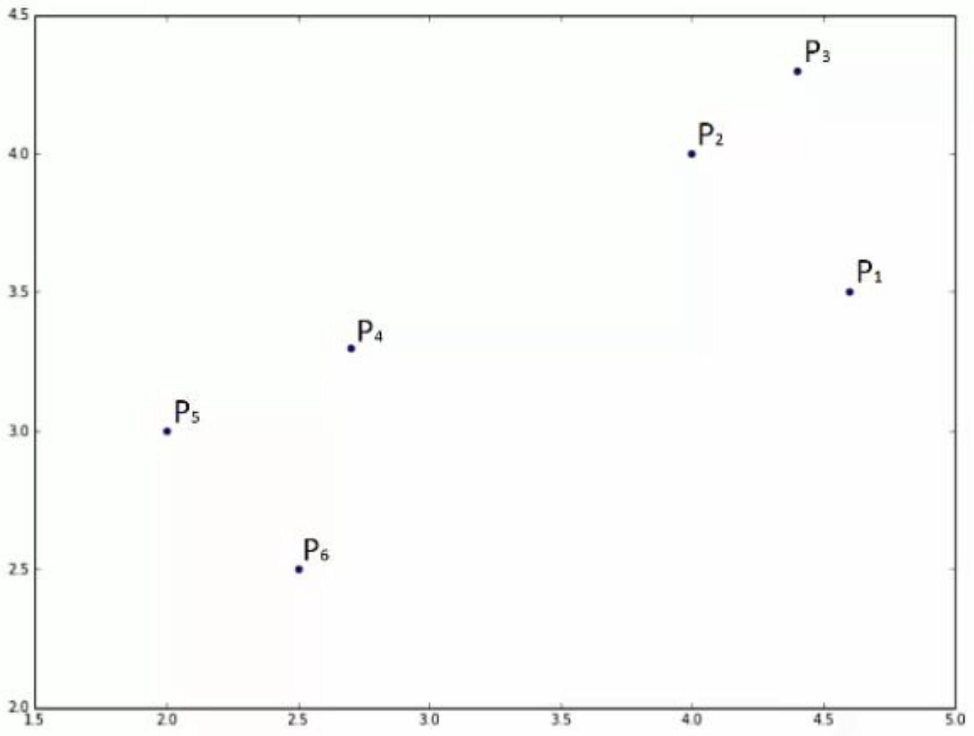
STEP 4: Repeat STEP 3 until there is only one cluster



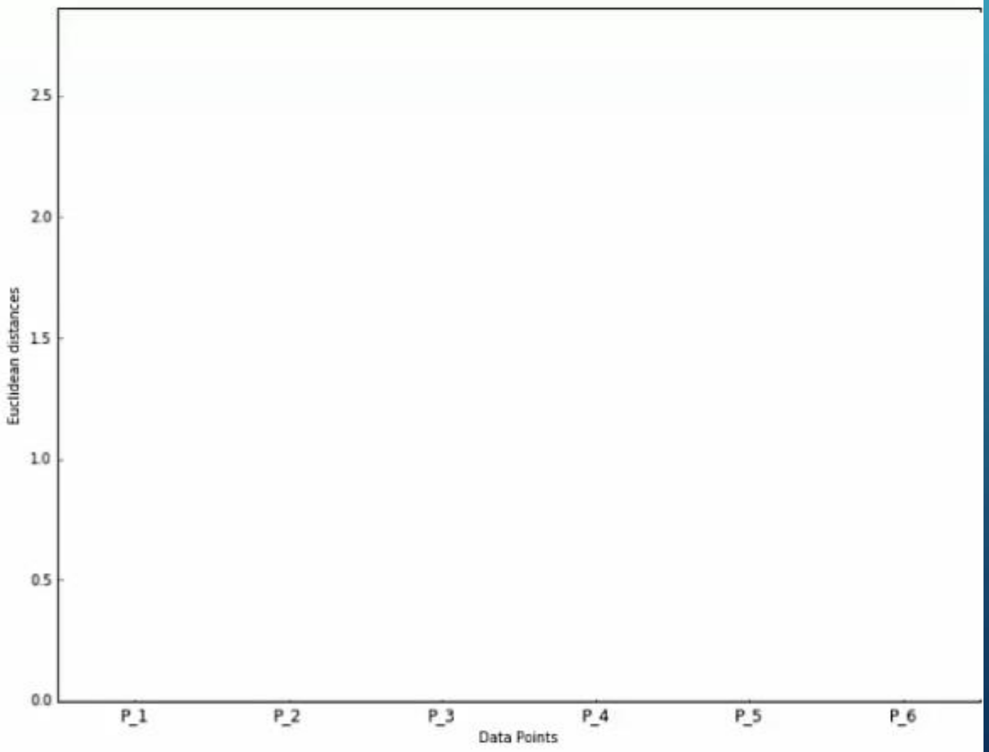

STEP 4: Repeat STEP 3 until there is only one cluster
FIN

DENDOGRAMS

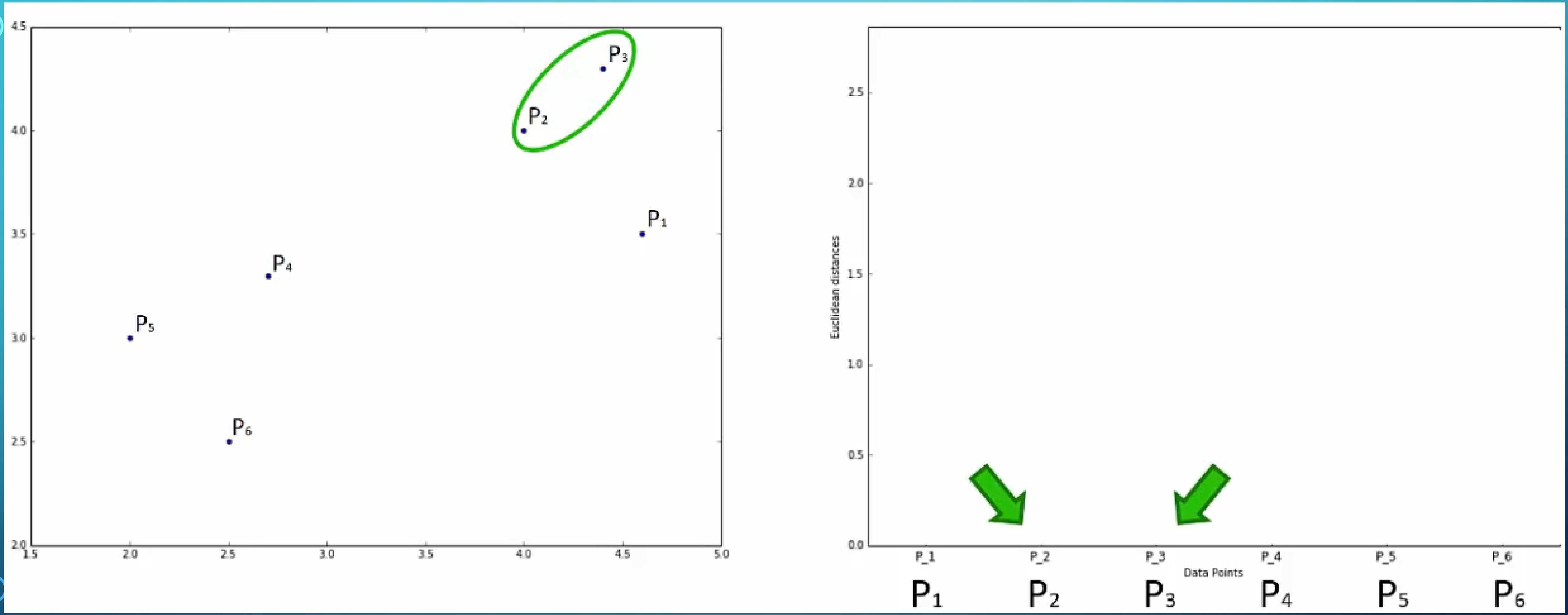
- How do they work??



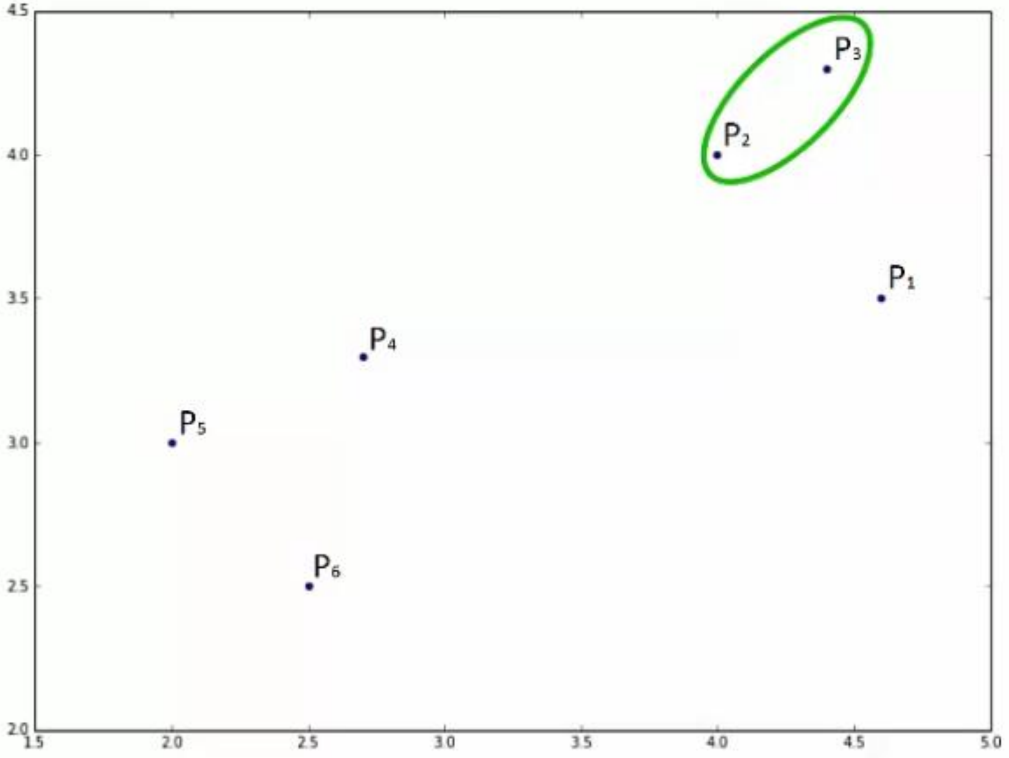
4.5
4.0
3.5
3.0
2.5
2.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
P_1
P_2
P_3
P_4
P_5
P_6



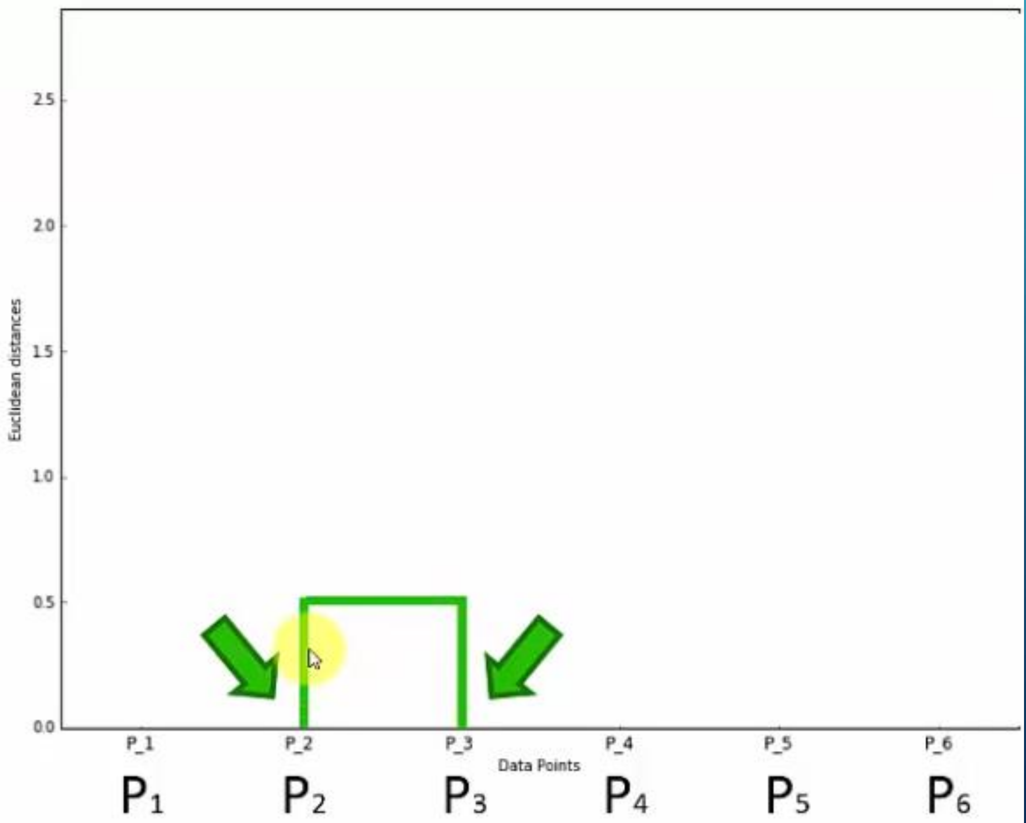
Euclidean distances
2.5
2.0
1.5
1.0
0.5
0.0
P_1
P_2
P_3
P_4
P_5
P_6
Data Points



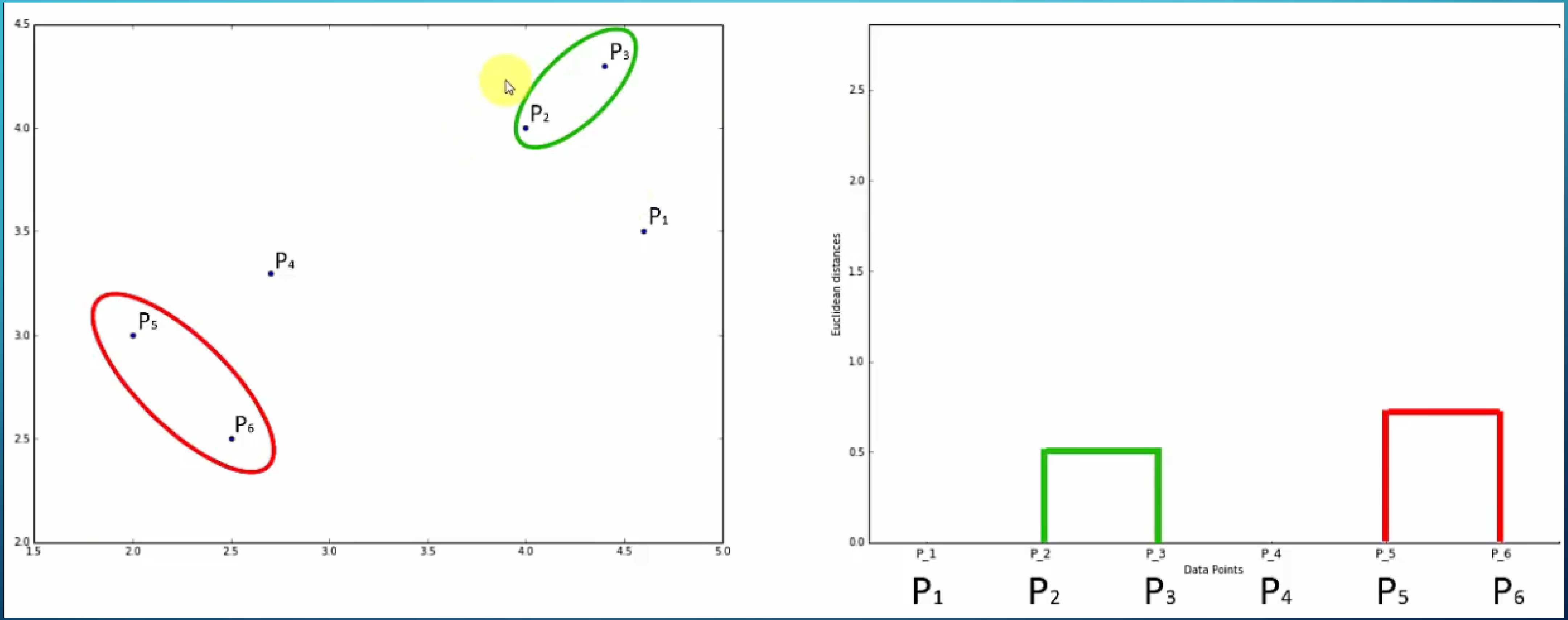
4.5
4.0
3.5
3.0
2.5
2.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
P_1
P_2
P_3
P_4
P_5
P_6
Euclidean distances
2.5
2.0
1.5
1.0
0.5
0.0
P_1
P_2
P_3
P_4
P_5
P_6
Data Points
P_1
P_2
P_3
P_4
P_5
P_6



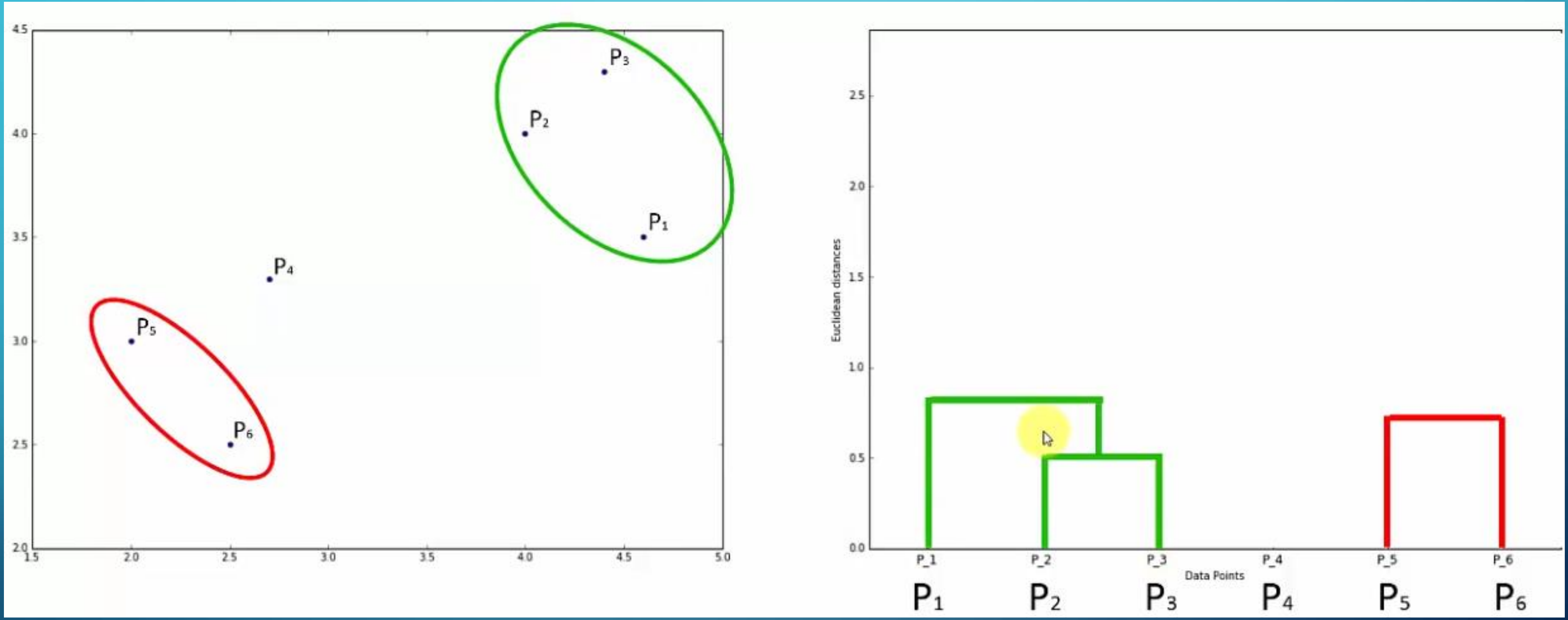
P1
P2
P3
P4
P5
P6



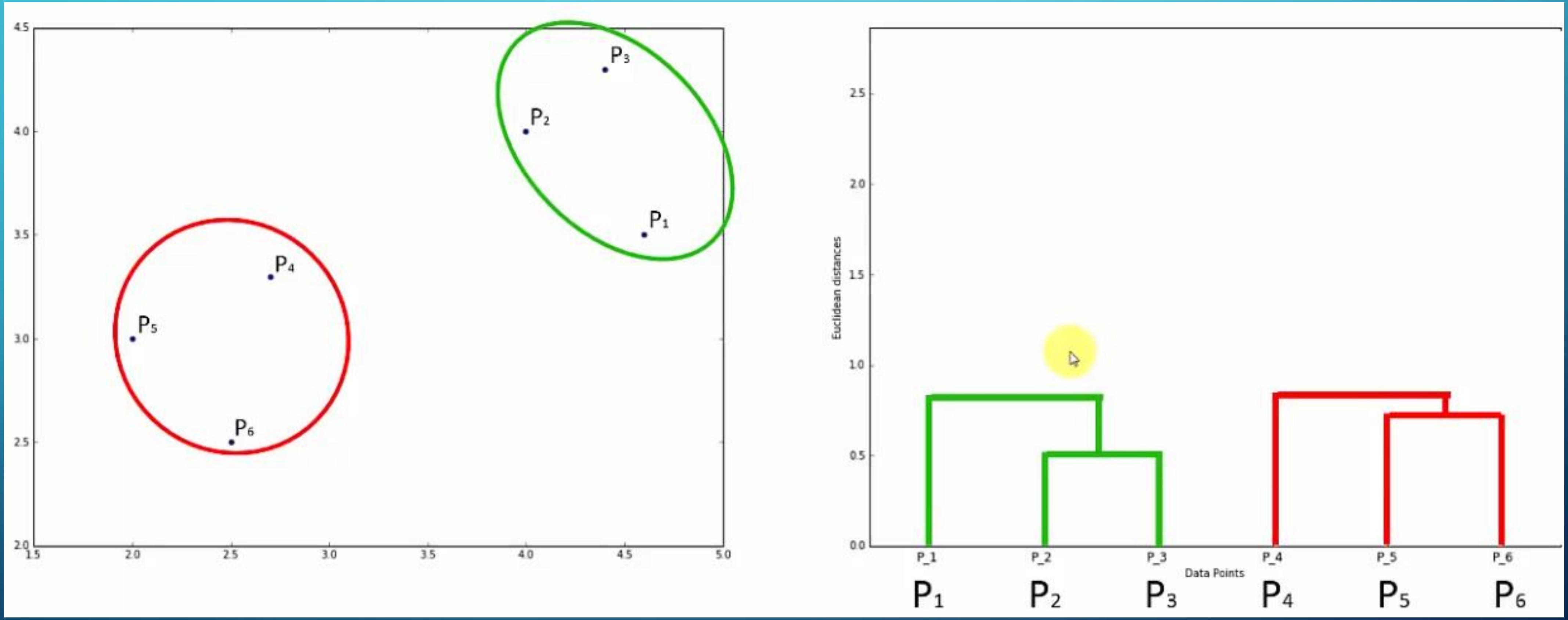
Euclidean distances
Data Points
P_1
P_2
P_3
P_4
P_5
P_6
P1
P2
P3
P4
P5
P6

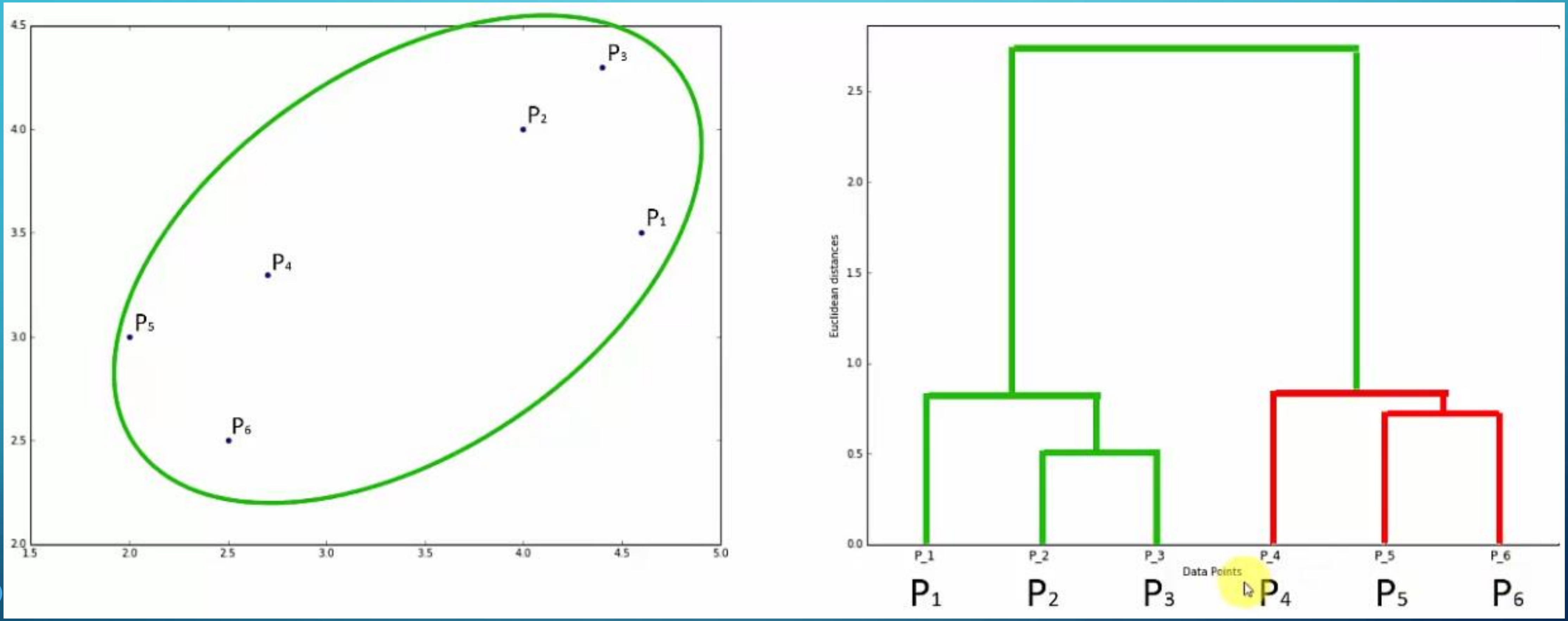


4.5
4.0
3.5
3.0
2.5
2.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
P1
P2
P3
P4
P5
P6
2.5
2.0
1.5
1.0
0.5
0.0
Euclidean distances
P_1
P_2
P_3
P_4
P_5
P_6
Data Points
P1
P2
P3
P4
P5
P6

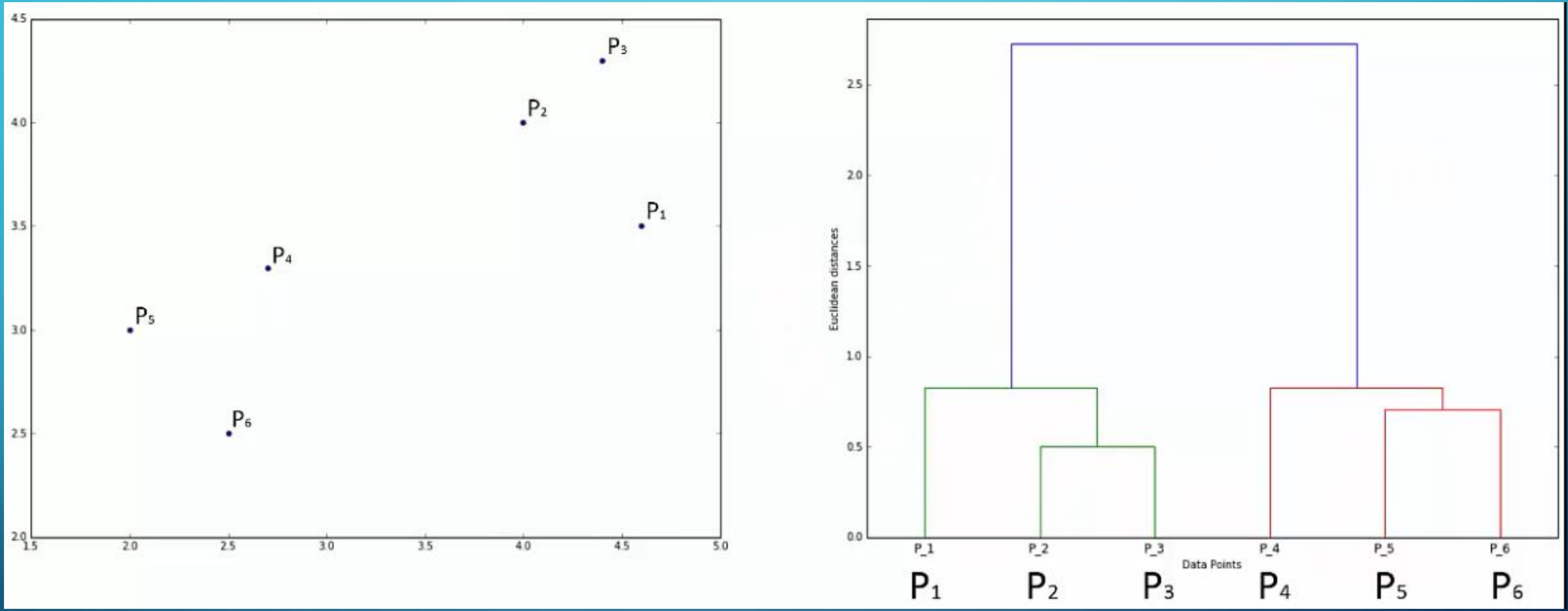


P3
P2
P1
P4
P5
P6
Euclidean distances
Data Points
P_1 P_2 P_3 P_4 P_5 P_6
P1 P2 P3 P4 P5 P6


P3
P2
P1
P4
P5
P6
4.5
4.0
3.5
3.0
2.5
2.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Euclidean distances
2.5
2.0
1.5
1.0
0.5
0.0
P_1
P_2
P_3
P_4
P_5
P_6
Data Points
P1
P2
P3
P4
P5
P6

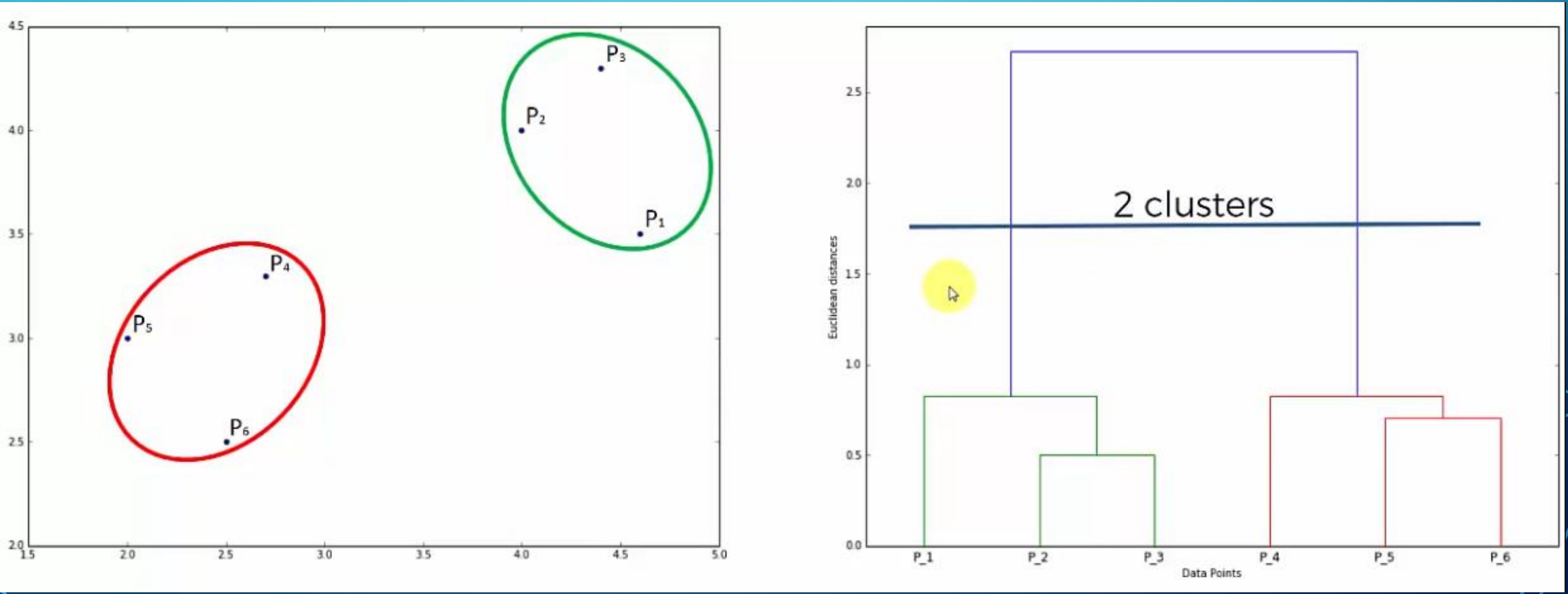


P₃
P₂
P₁
P₄
P₅
P₆
Euclidean distances
Data Points
P_1
P_2
P_3
P_4
P_5
P_6

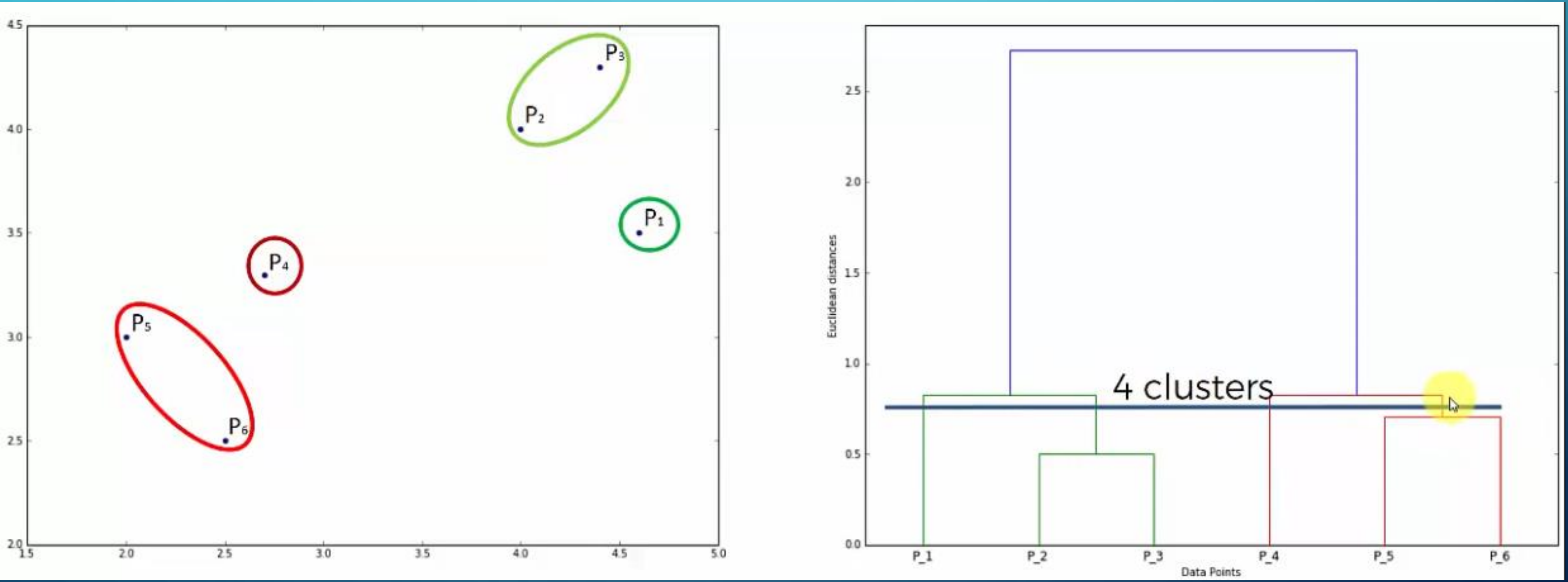


4.5
4.0
3.5
3.0
2.5
2.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
P1
P2
P3
P4
P5
P6
Euclidean distances
0.0
0.5
1.0
1.5
2.0
2.5
P_1
P_2
P_3
P_4
P_5
P_6
Data Points
P1
P2
P3
P4
P5
P6

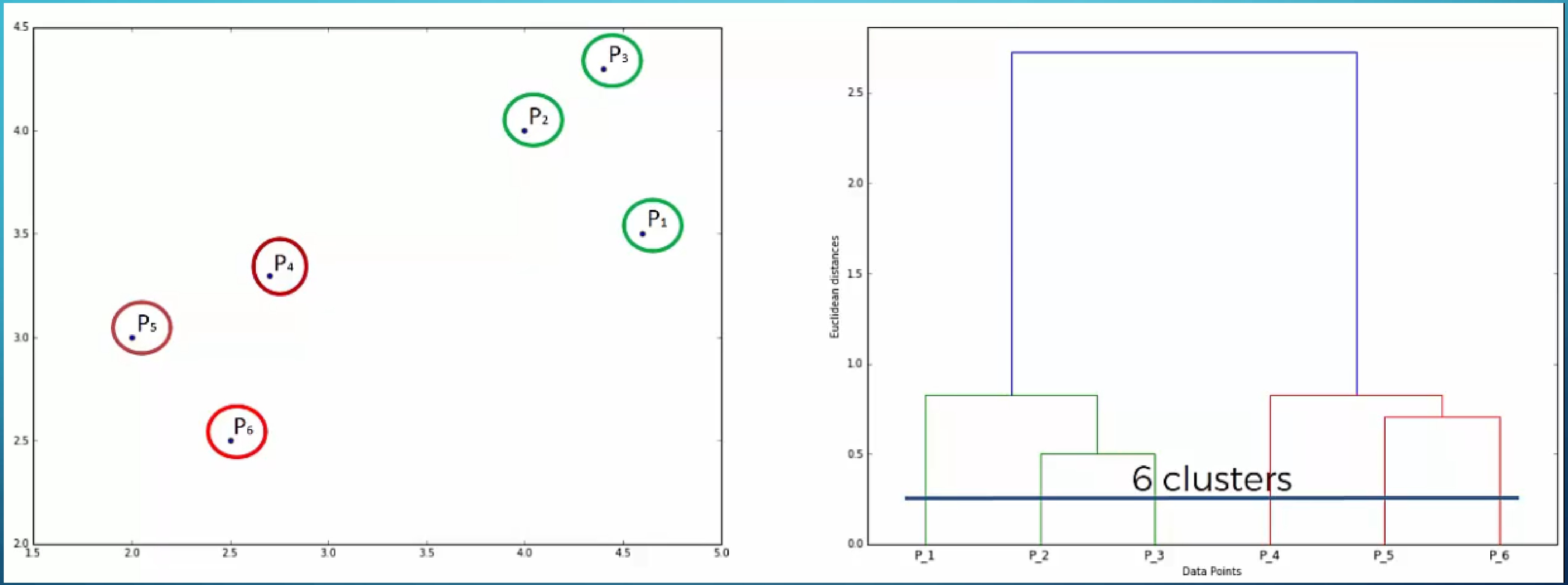
DENDOGRAMS: 2-CLUSTERS



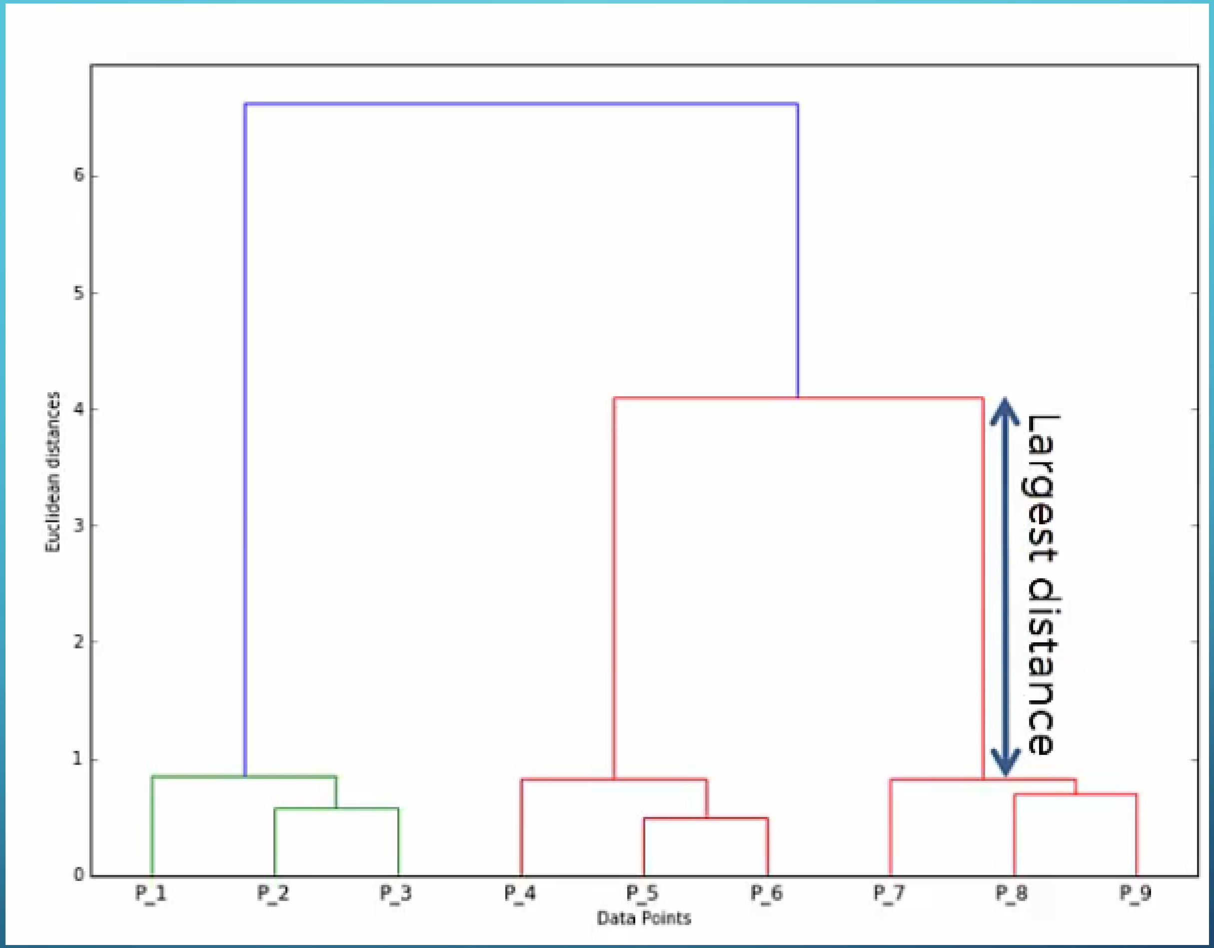
DENDOGRAMS: 4-CLUSTERS

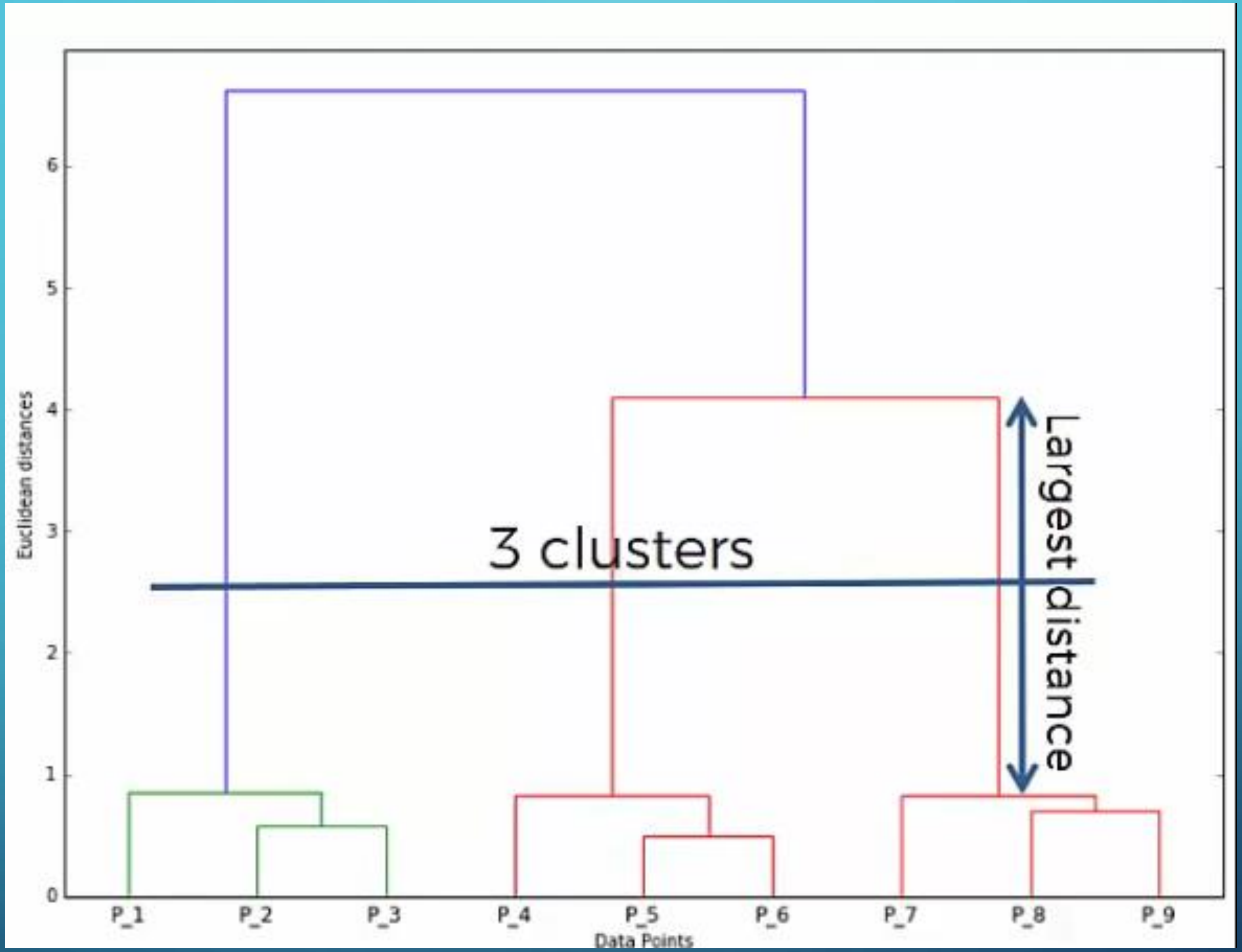


DENDOGRAMS: 6-CLUSTERS

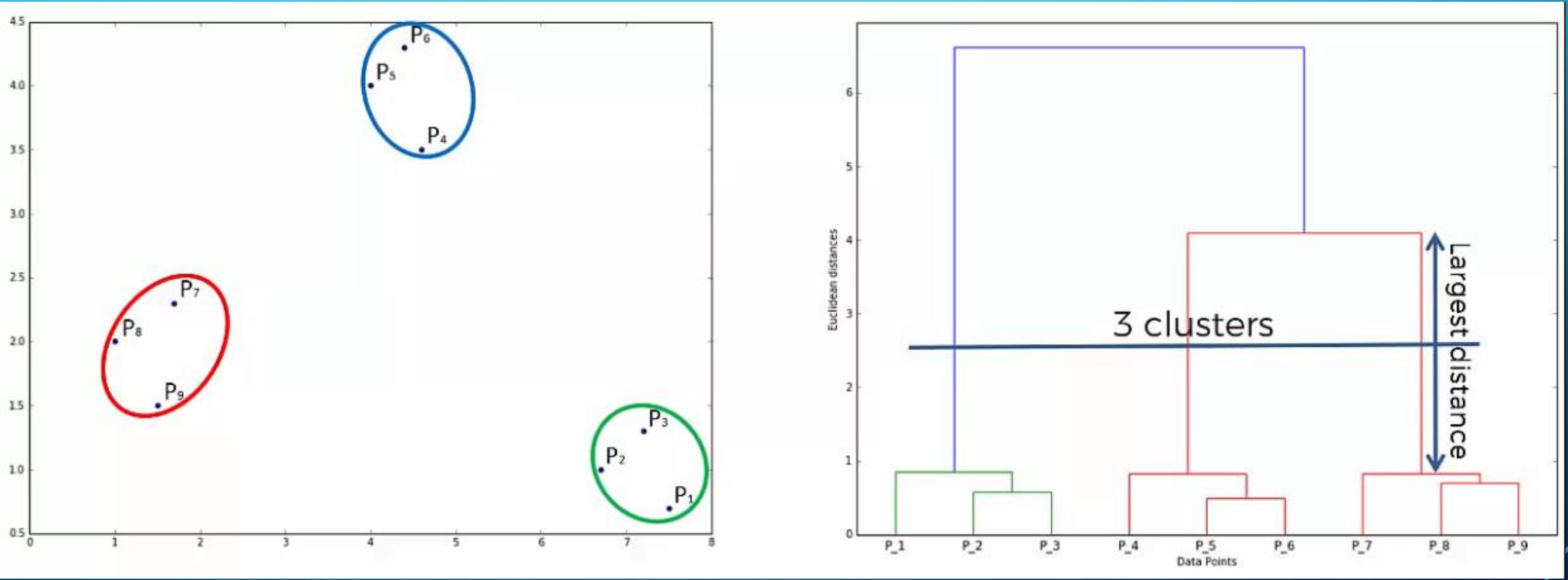


WHAT IS OPTIMAL # OF CLUSTERS??


Euclidean distances
0
1
2
3
4
5
6
P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9
Data Points
Largest distance


3 clusters
Largest distance
Euclidean distances
Data Points
0
1
2
3
4
5
6
P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9

OPTIMAL 3-CLUSTERS



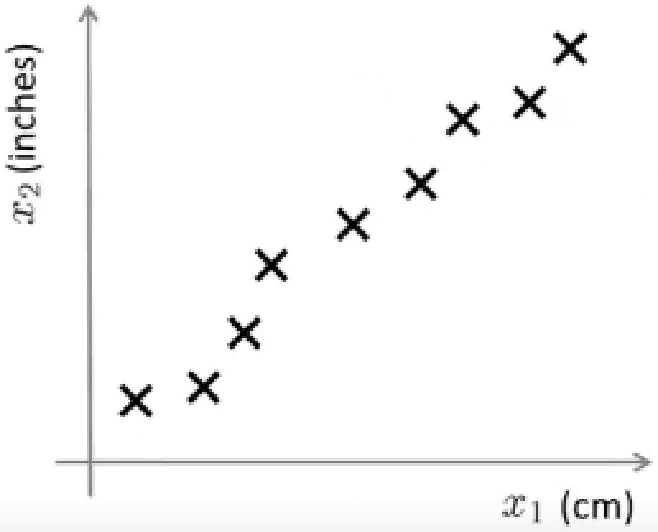
PUT ON YOUR THINKING HAT!!

Its time to implement!!

PRINCIPLE COMPONENT ANALYSIS

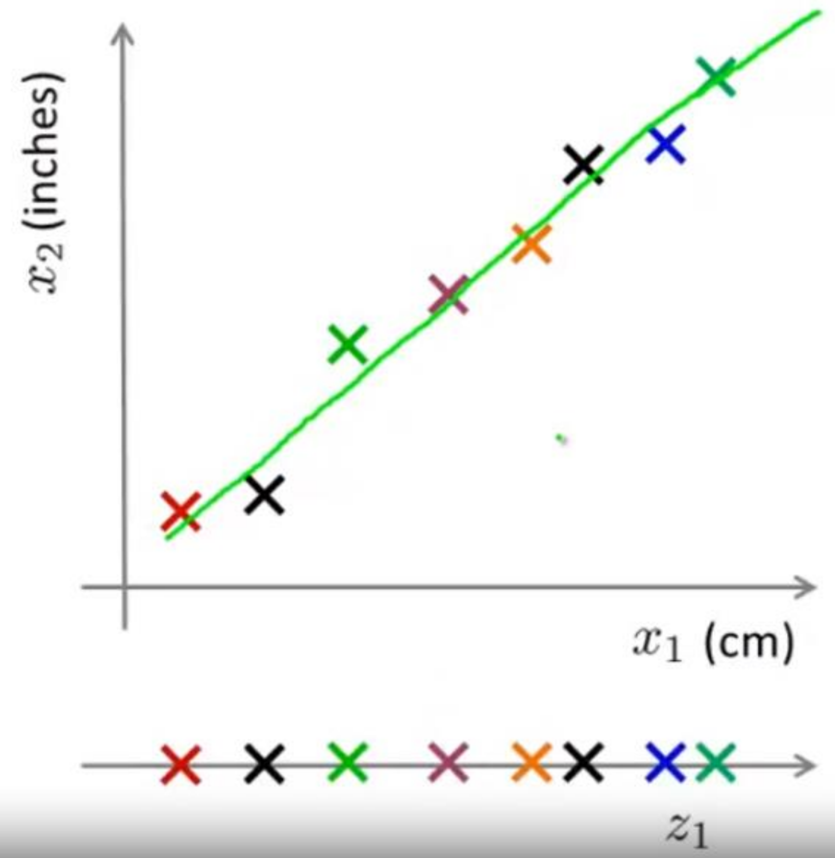
- Dimensionality Reduction
- Data Compression
- Helps in Data Visualisation

Data Compression



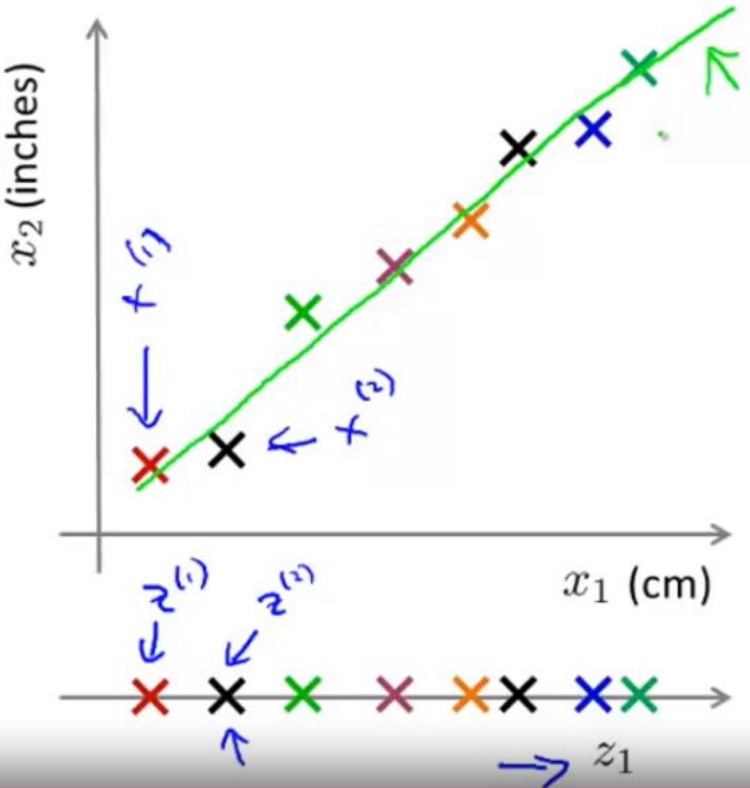
Reduce data from 2D to 1D

Data Compression



Reduce data from 2D to 1D

Data Compression



Reduce data from 2D to 1D

$x^{(1)} \in \mathbb{R}^2 \quad \rightarrow z^{(1)} \in \mathbb{R}$

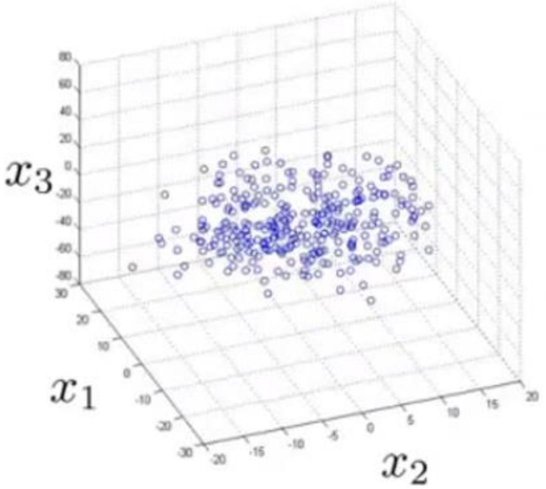
$x^{(2)} \in \mathbb{R}^2 \quad \rightarrow z^{(2)} \in \mathbb{R}$

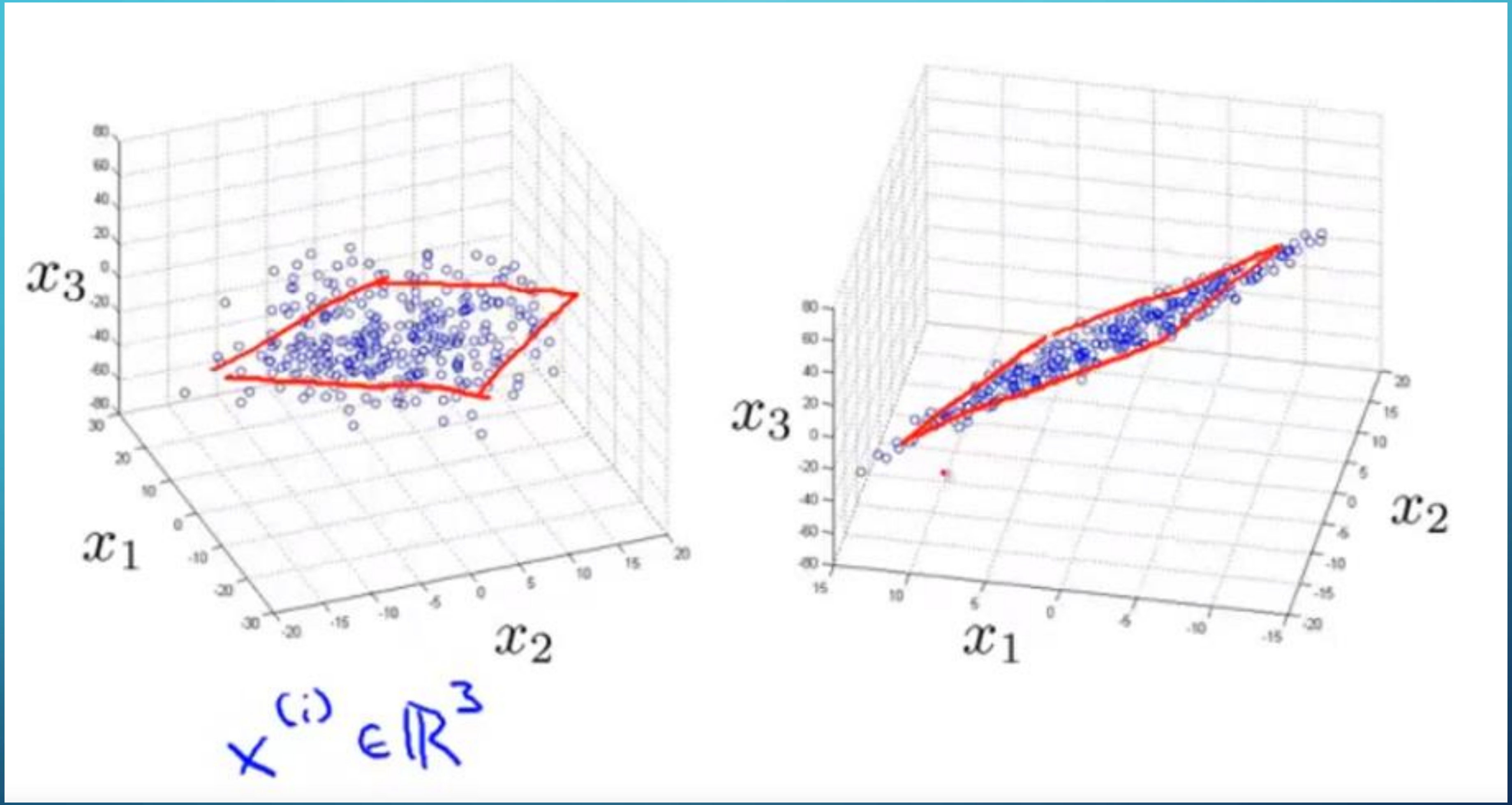
$\vdots$

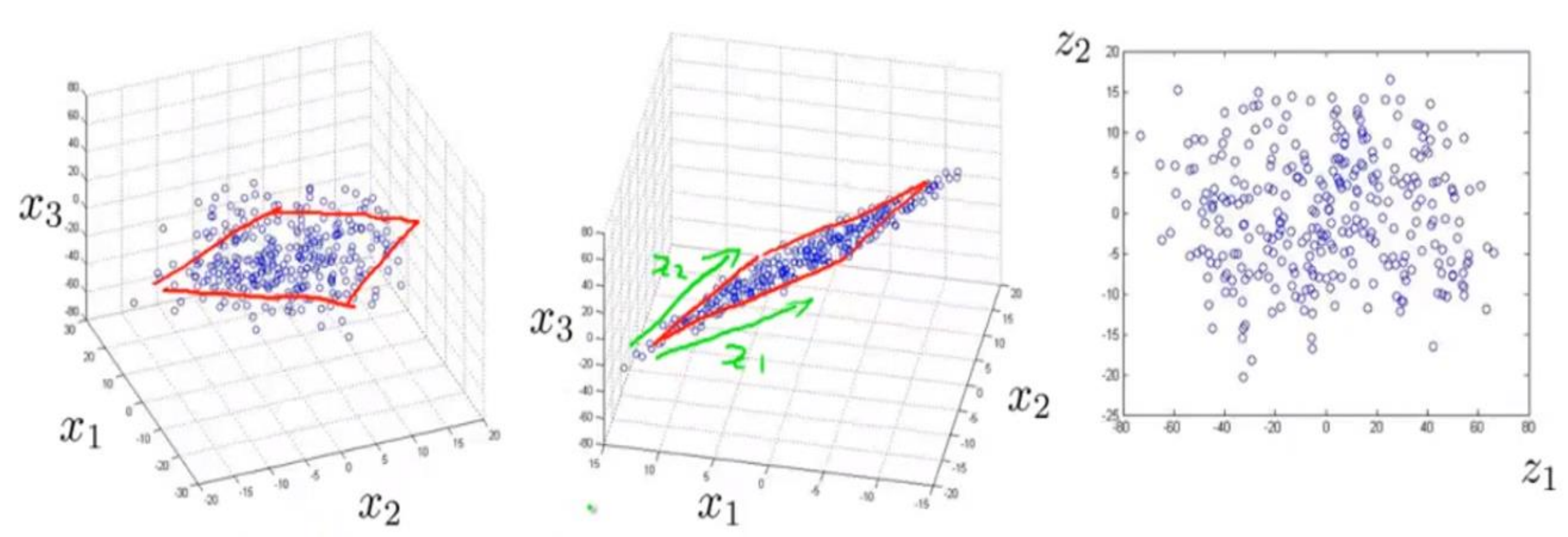
$x^{(m)} \quad \rightarrow z^{(m)}$

Data Compression

Reduce data from 3D to 2D




x_3
x_1
x_2
$x^{(i)} \in \mathbb{R}^3$
x_3
x_1
x_2


x_3
x_1
x_2
z_2
z_1
x_3
x_1
x_2
z_2
z_1

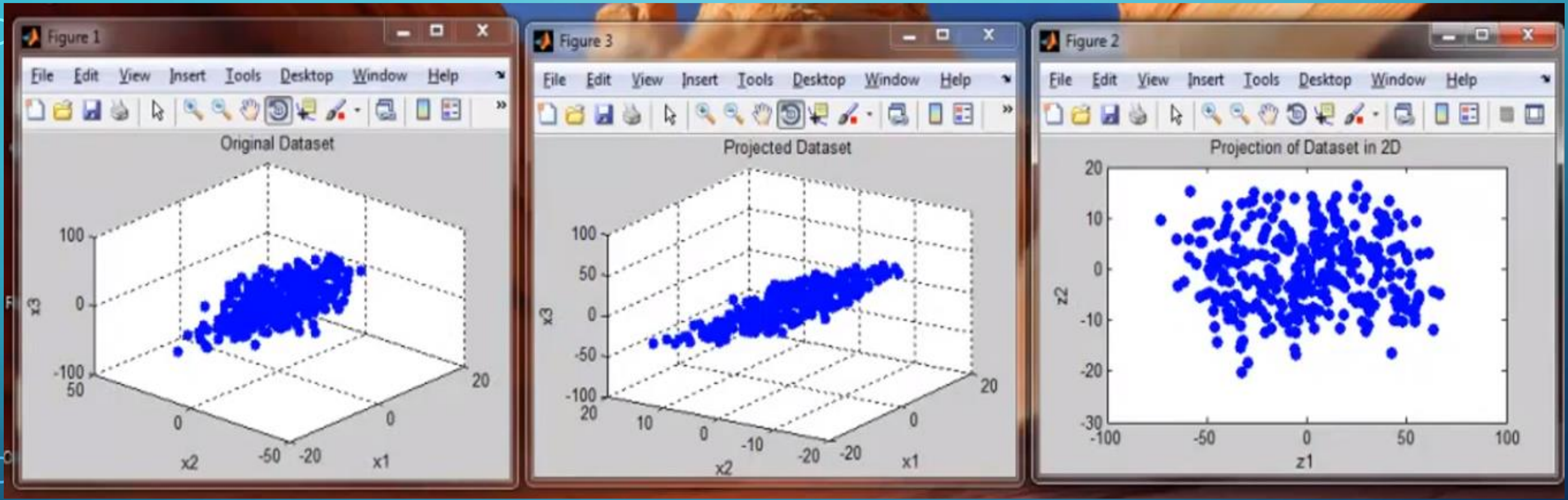
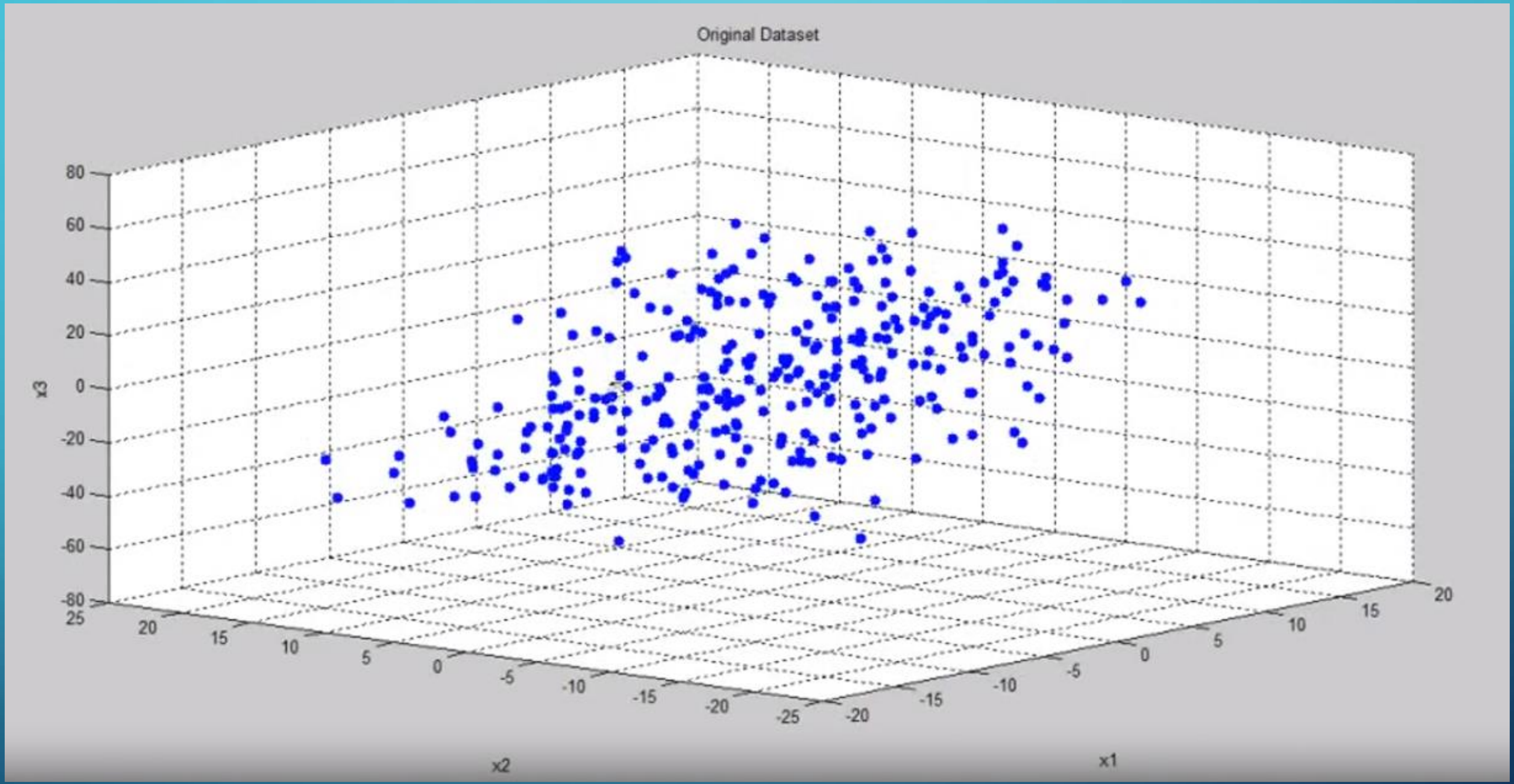
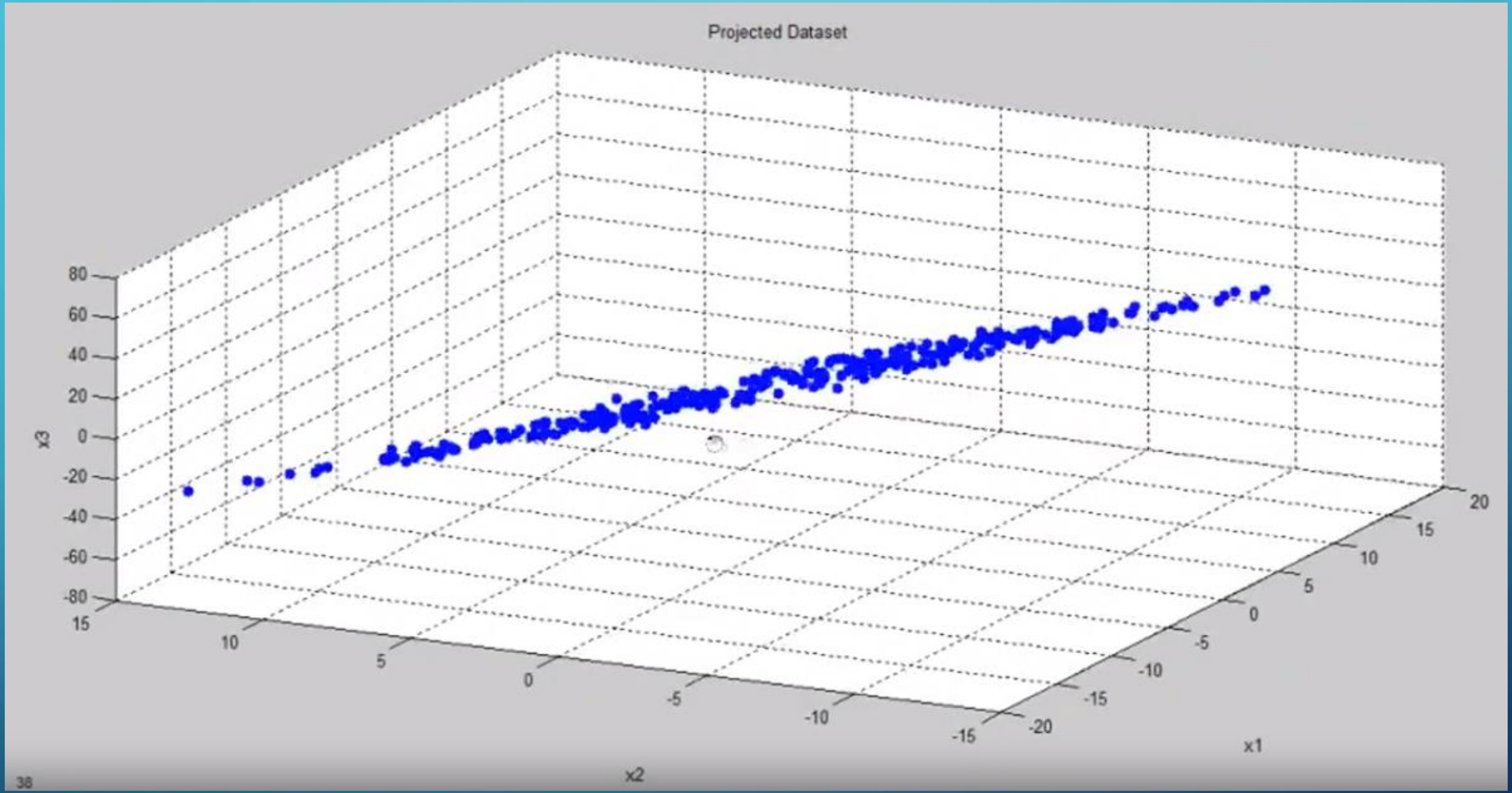


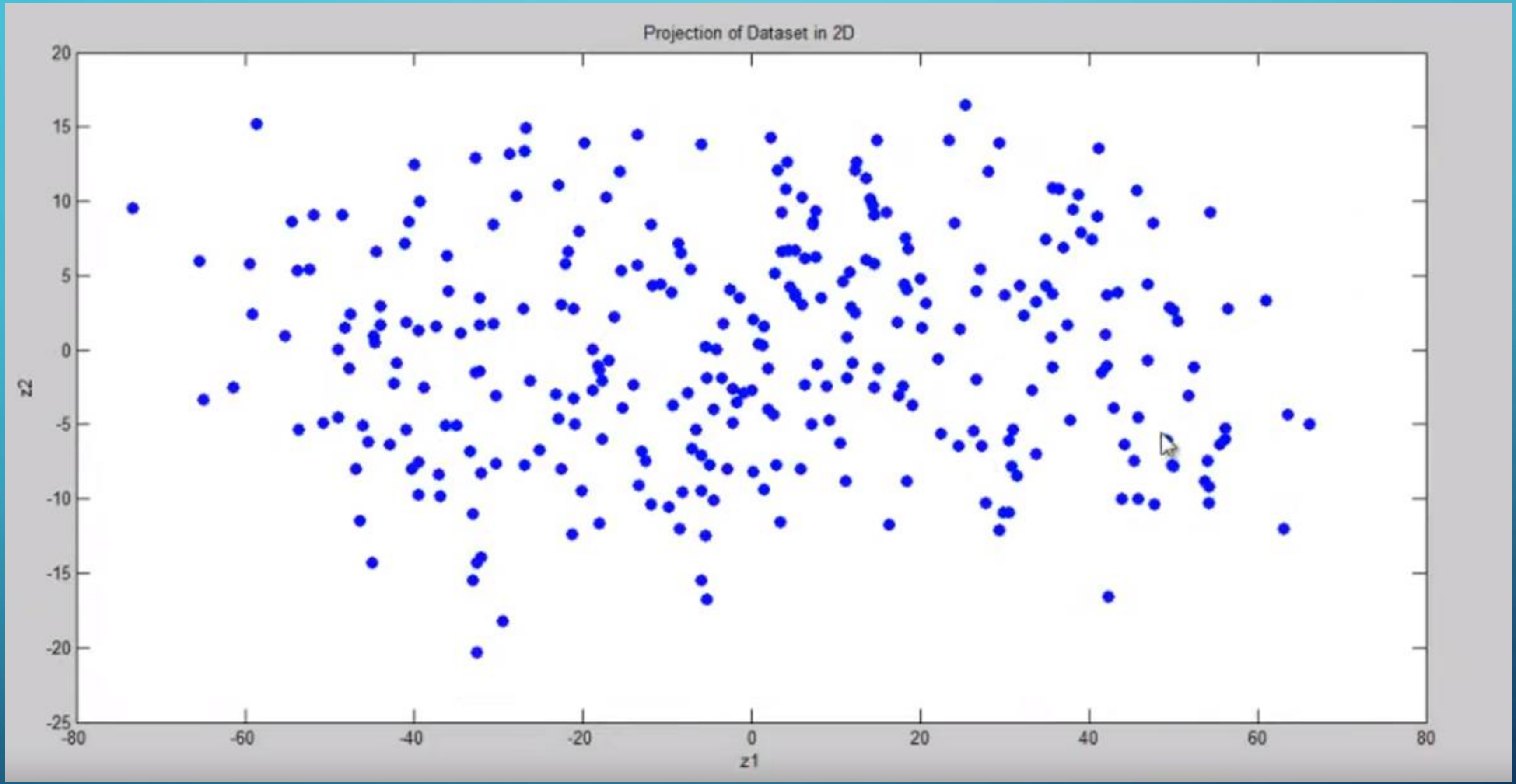
Figure 1
File Edit View Insert Tools Desktop Window Help
Original Dataset
x3
x2
x1
Figure 3
File Edit View Insert Tools Desktop Window Help
Projected Dataset
x3
x2
x1
Figure 2
File Edit View Insert Tools Desktop Window Help
Projection of Dataset in 2D
z2
z1



Original Dataset
x3
80
60
40
20
0
-20
-40
-60
-80
x2
25
20
15
10
5
0
-5
-10
-15
-20
-25
x1
-20
-15
-10
-5
0
5
10
15
20



Projected Dataset
x3
80
60
40
20
0
-20
-40
-60
-80
15
10
5
0
-5
-10
-15
x2
20
15
10
5
0
-5
-10
-15
-20
x1
38



Projection of Dataset in 2D
z2
z1
20
15
10
5
0
-5
-10
-15
-20
-25
-80
-60
-40
-20
0
20
40
60
80

Data Visualization

$x \in \mathbb{R}^{50}$

Country	x_1 GDP (trillions of US$)	x_2 Per capita GDP (thousands of intl. $)	x_3 Human Develop-ment Index	x_4 Life expectancy	x_5 Poverty Index (Gini as percentage)	x_6 Mean household income (thousands of US$)	…
Canada	1.577	39.17	0.908	80.7	32.6	67.293	…
China	5.878	7.54	0.687	73	46.9	10.22	…
India	1.632	3.41	0.547	64.7	36.8	0.735	…
Russia	1.48	19.84	0.755	65.5	39.9	0.72	…
Singapore	0.223	56.69	0.866	80	42.5	67.1	…
USA	14.527	46.86	0.91	78.3	40.8	84.3	…
…	…	…	…	…	…	…	

DATA VISUALISATION NOT POSSIBLE FOR 50D

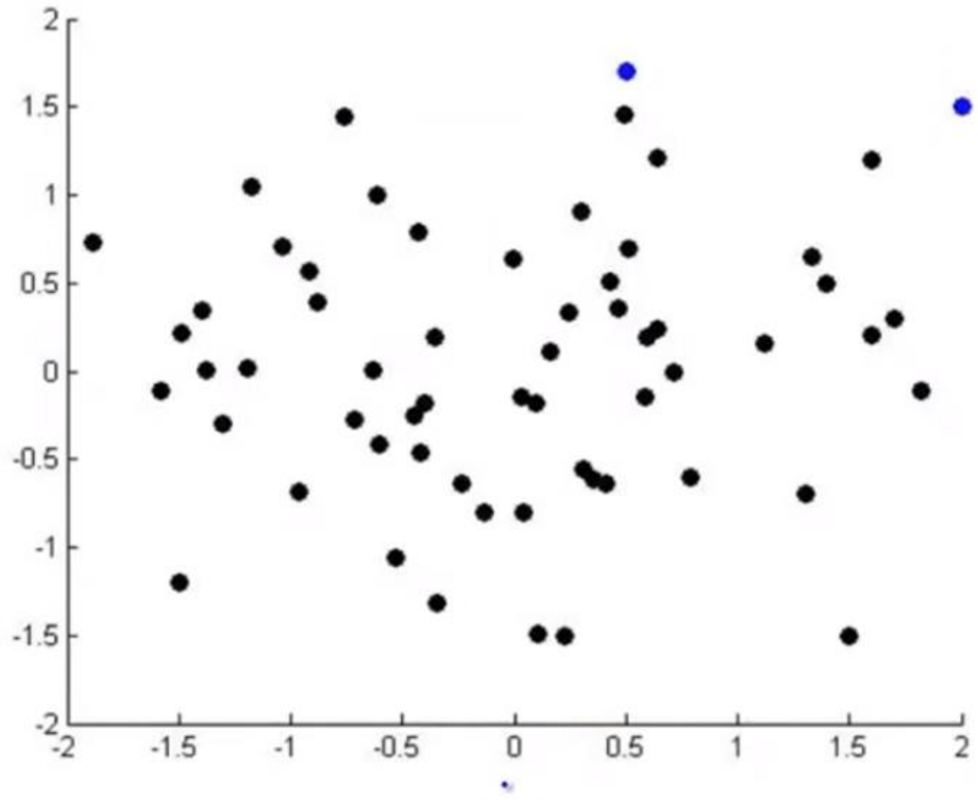
Data Visualization

Country	z_1	z_2
Canada	1.6	1.2
China	1.7	0.3
India	1.6	0.2
Russia	1.4	0.5
Singapore	0.5	1.7
USA	2	1.5
...	...	...

$z^{(i)} \in \mathbb{R}^2$

Reduce data from 50D to 2D

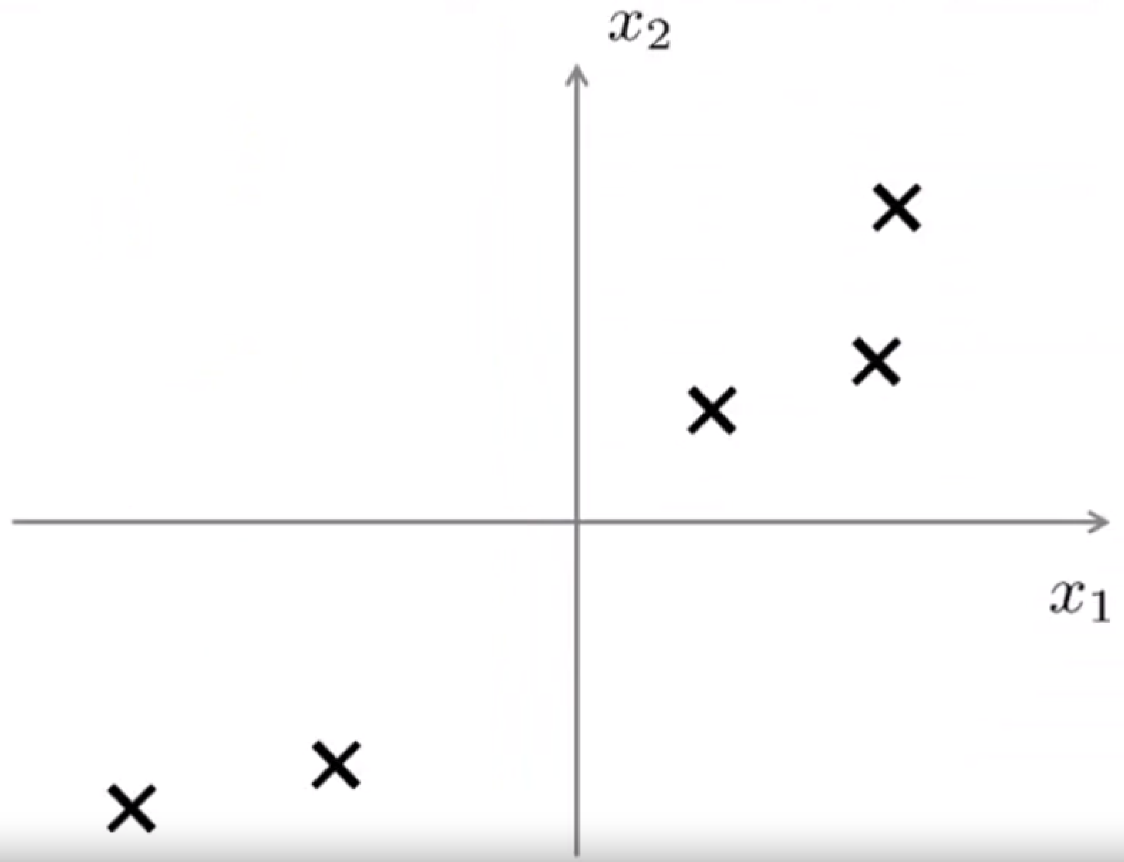
Data Visualization



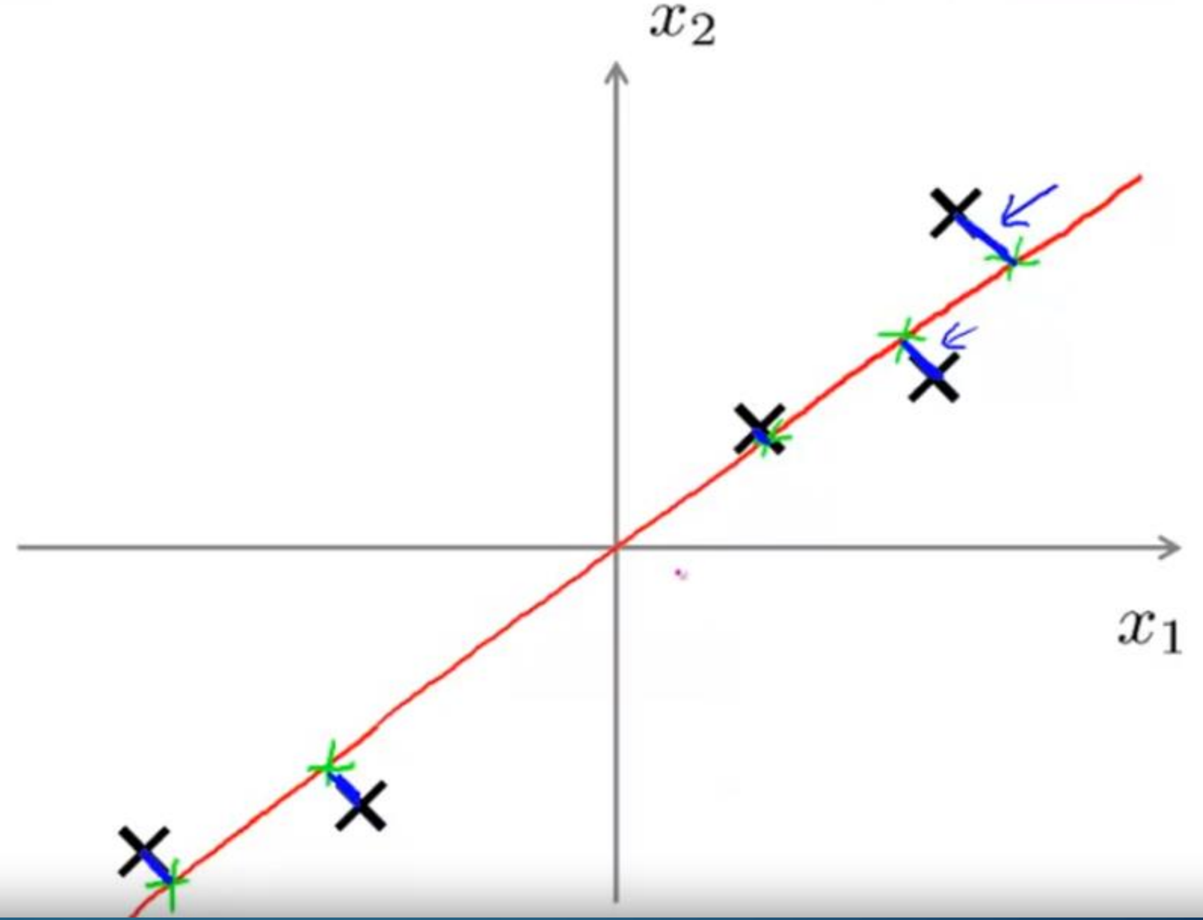
1

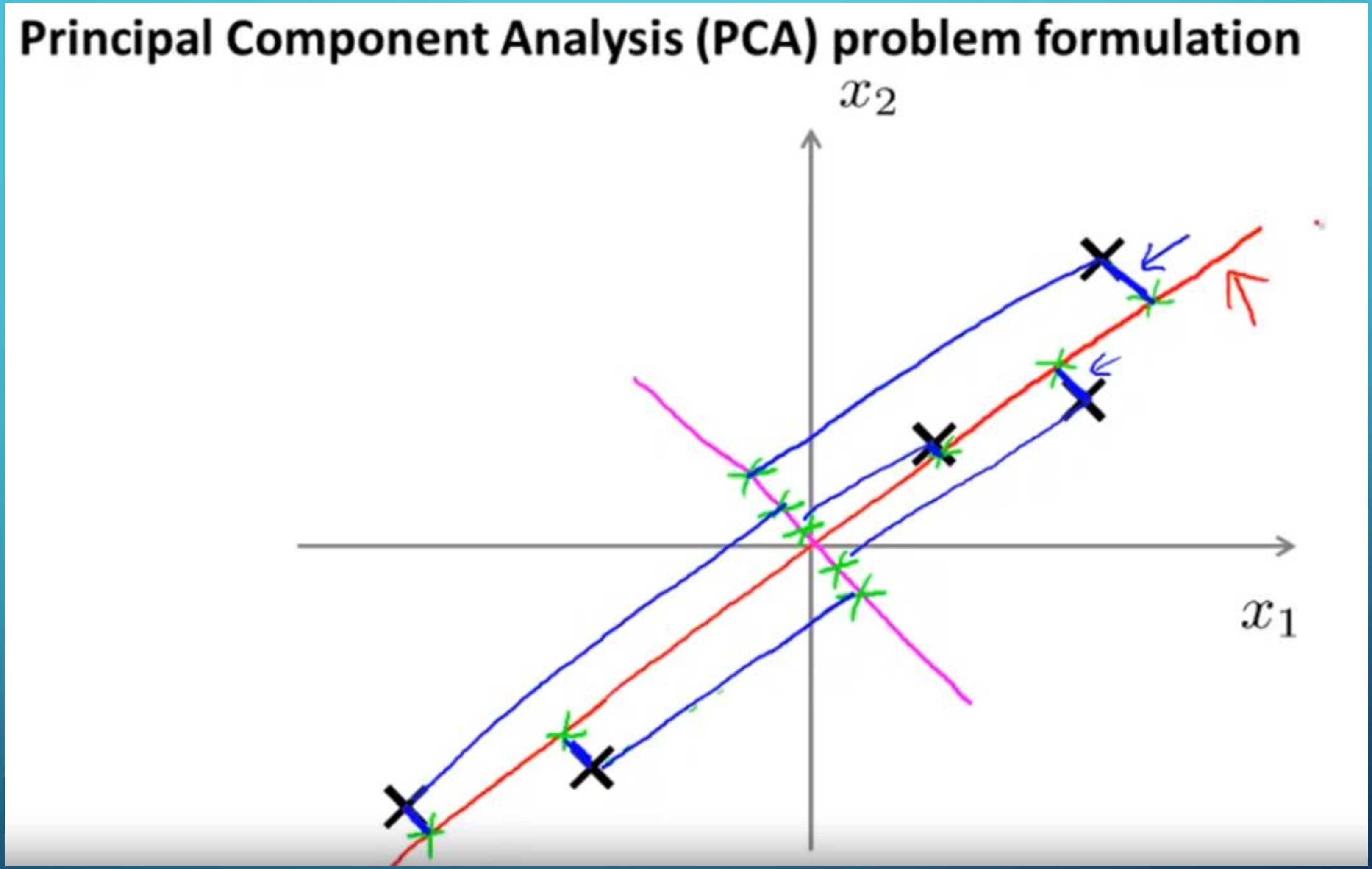
PCA PROBLEM FORMULATION

Principal Component Analysis (PCA) problem formulation



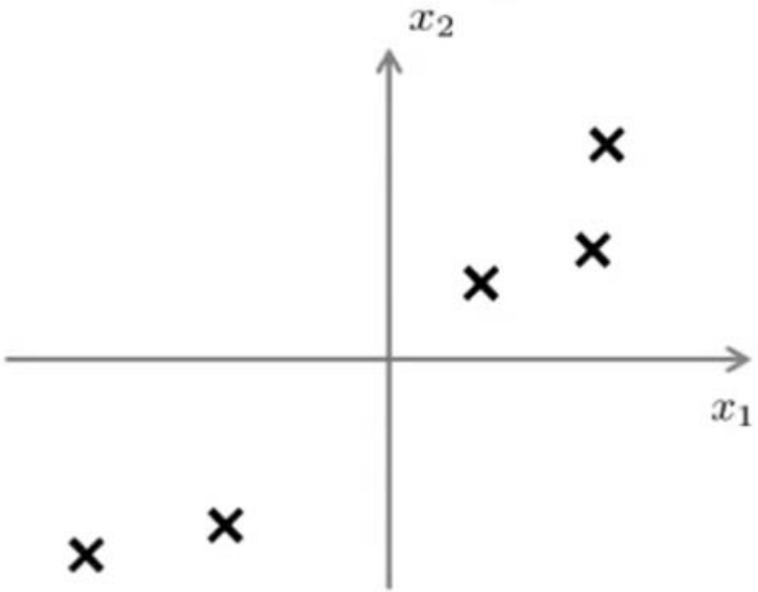
Principal Component Analysis (PCA) problem formulation





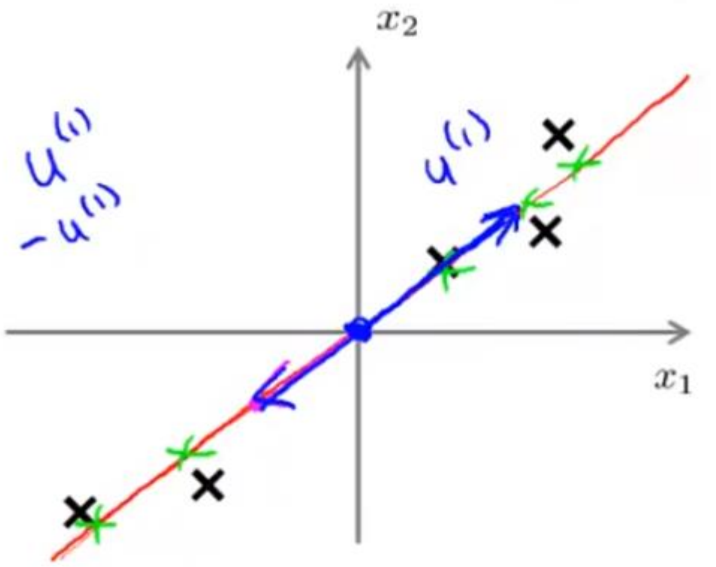
Principal Component Analysis (PCA) problem formulation
x_2
x_1

Principal Component Analysis (PCA) problem formulation



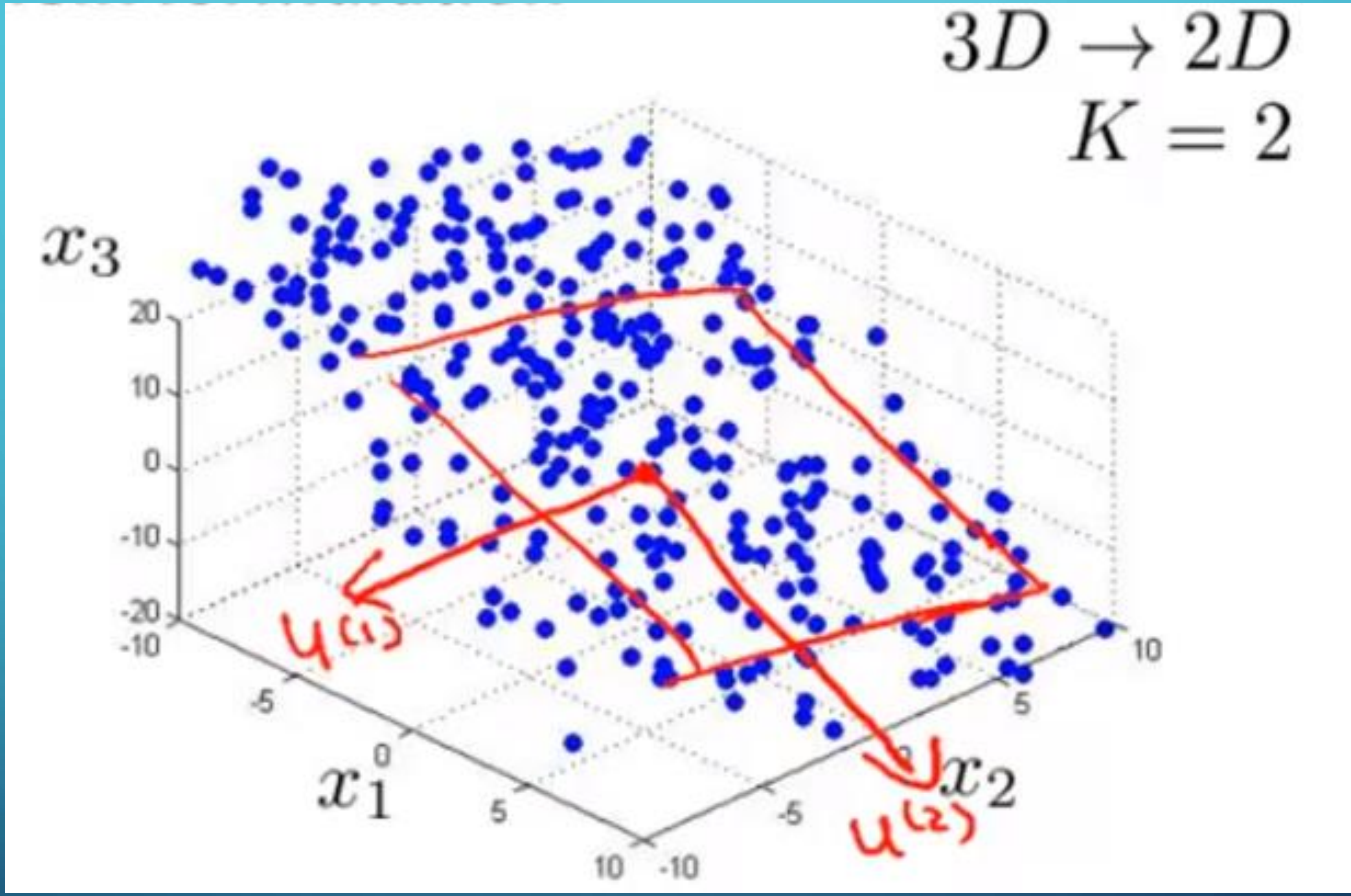
Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Principal Component Analysis (PCA) problem formulation



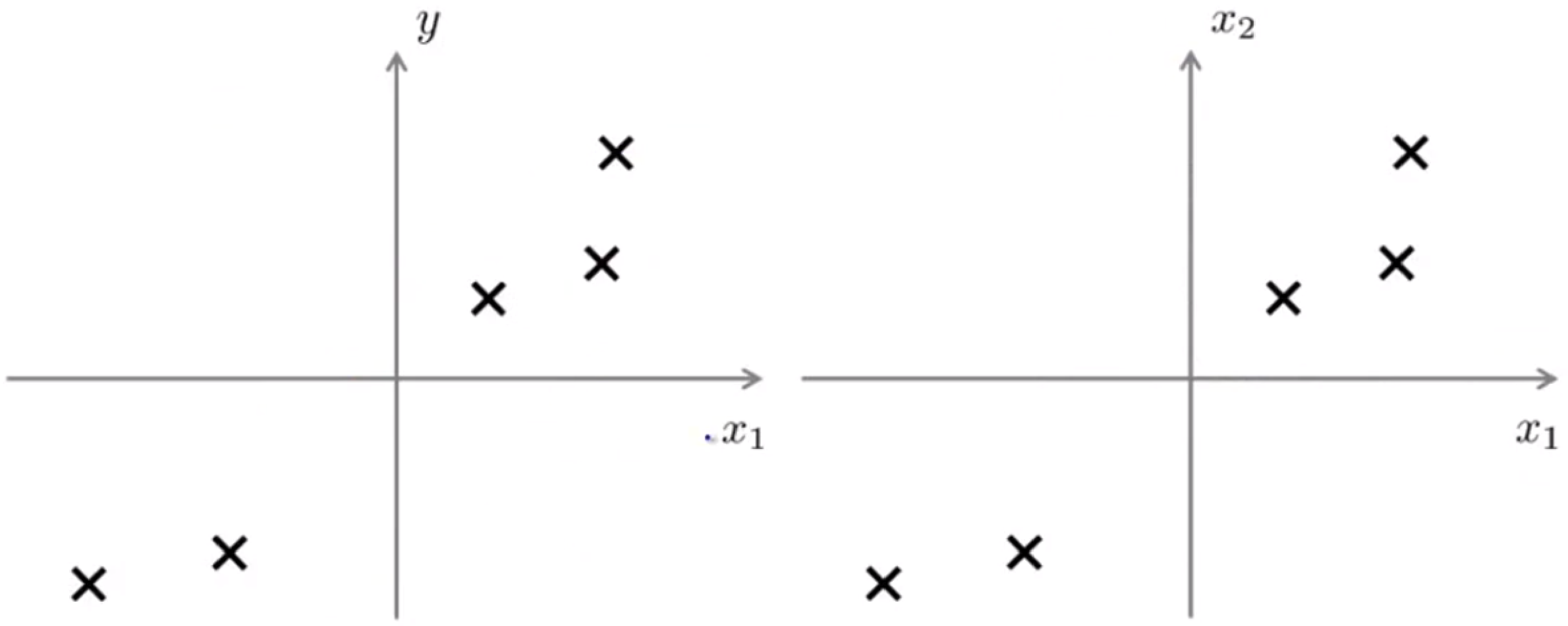
Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n-dimension to k-dimension: Find k vectors $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

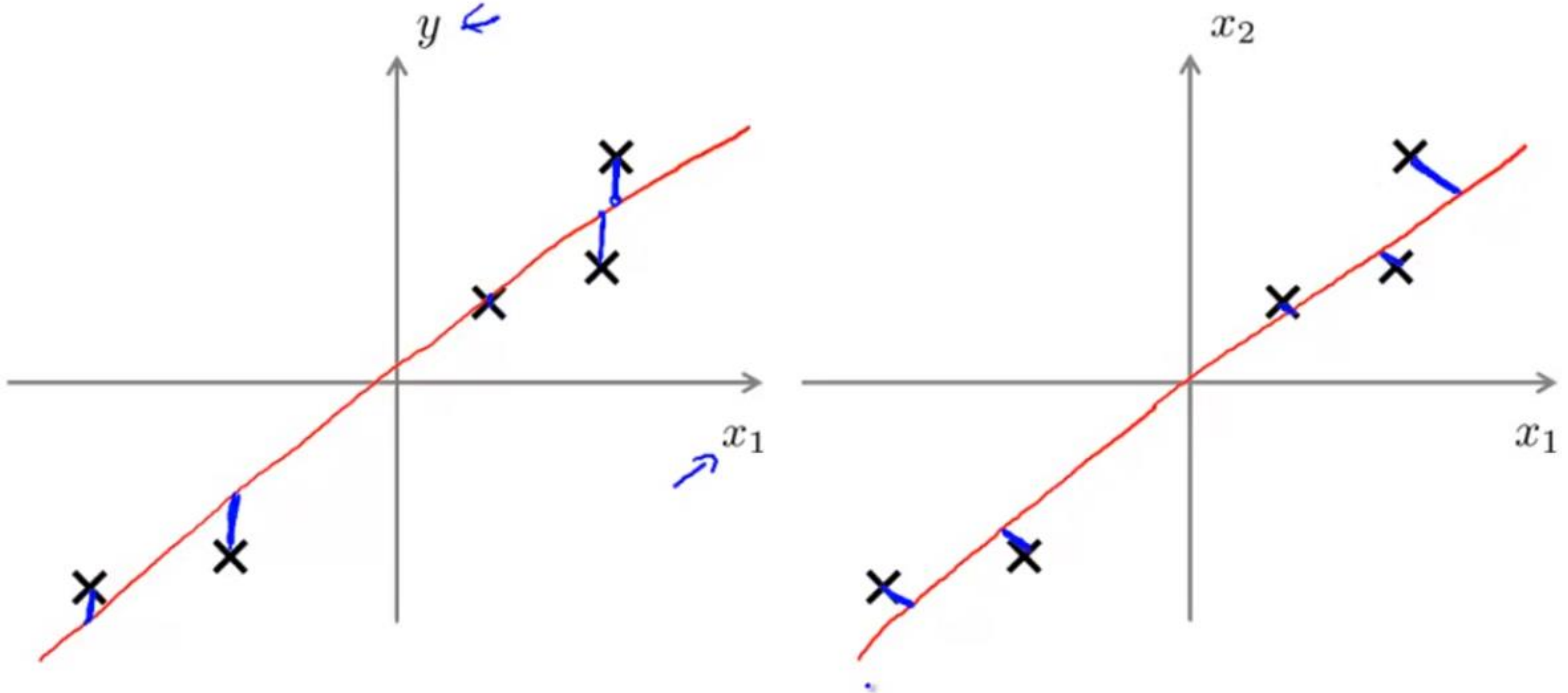


$3D \rightarrow 2D$
$K = 2$
x_3
20
10
0
-10
-20
-10
-5
0
5
10
x_1
-10
-5
5
10
x_2
$u^{(1)}$
$u^{(2)}$

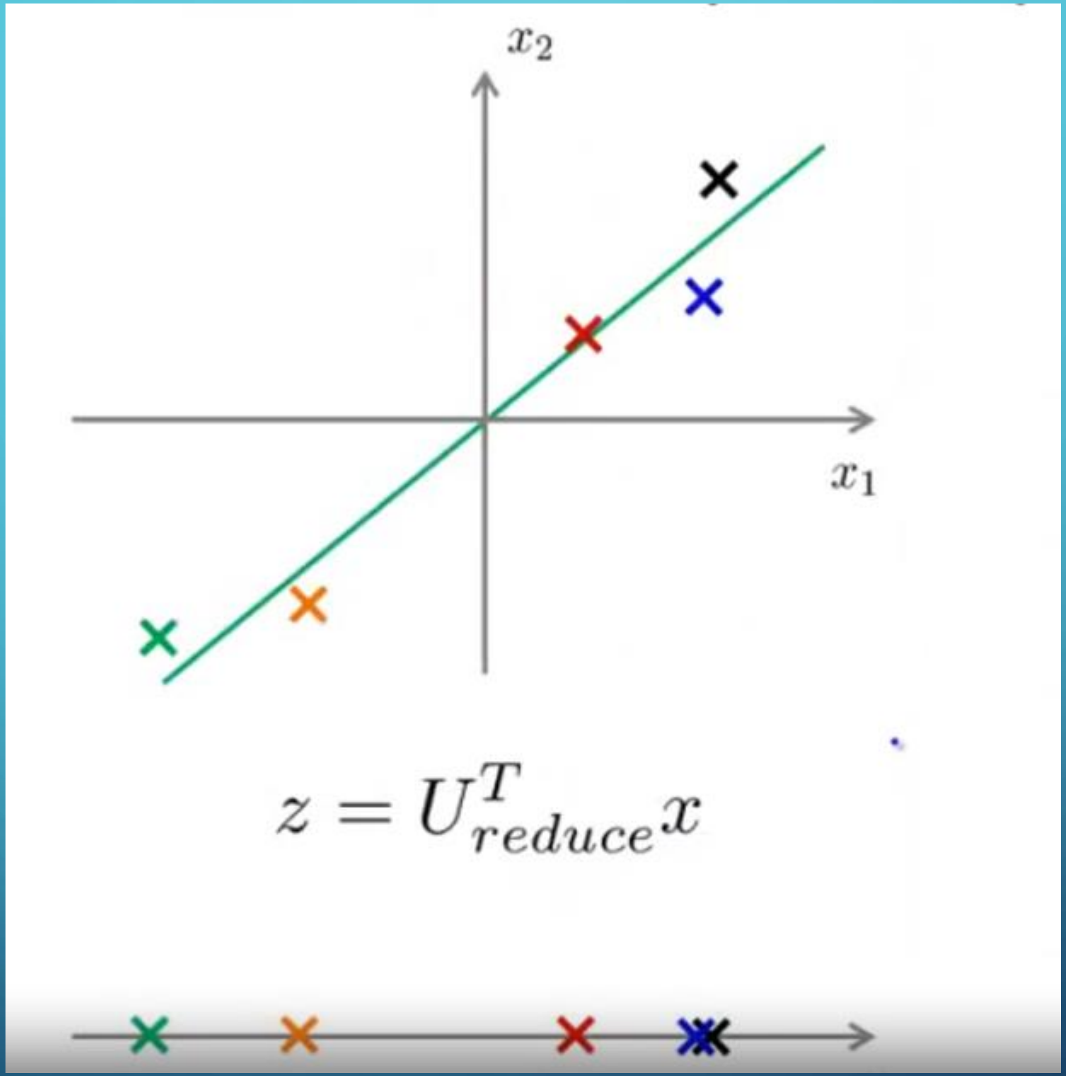
PCA is not linear regression



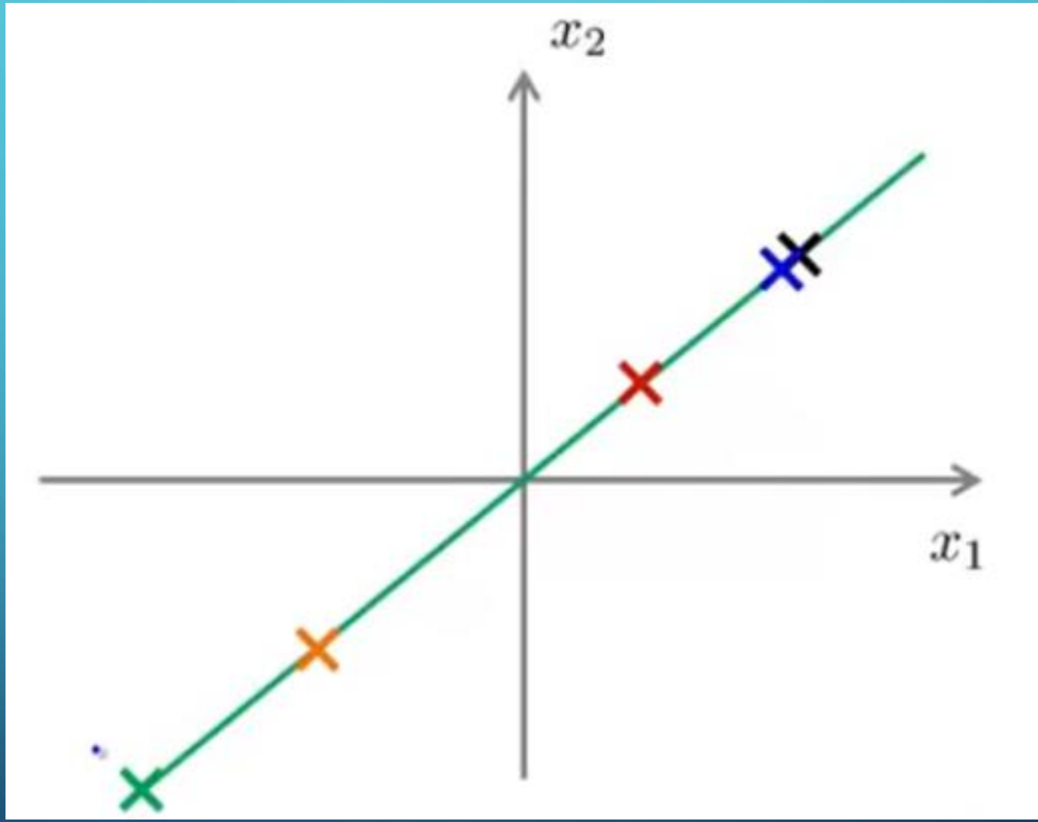
PCA is not linear regression



RECONSTRUCTION FROM COMPRESSED REPRESENTATION


x_2
x_1
$z = U_{reduce}^T x$

VARIANCE IN DATA IS LOST



Application of PCA

- Compression
  - Reduce memory/disk needed to store data
  - Speed up learning algorithm
- Visualization

Bad use of PCA: To prevent overfitting

Use $z^{(i)}$ instead of $x^{(i)}$ to reduce the number of features to $k < n$.

Thus, fewer features, less likely to overfit.

This might work OK, but isn't a good way to address overfitting. Use regularization instead.

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

TIME TO CODE PCA!!

APRIORI


Pampers
Swaddlers
New Baby
1
8-14 lb
4-6 kg
20
DIAPERS
COUCHES
PAÑALES


Heineken
HEINEKEN LAGER BEER
PREMIUM QUALITY

BASIC IDEA

People who bought also bought ...

MOVIE RECOMMENDATION

User ID	Movies liked
46578	Movie1, Movie2, Movie3, Movie4
98989	Movie1, Movie2
71527	Movie1, Movie2, Movie4
78981	Movie1, Movie2
89192	Movie2, Movie4
61557	Movie1, Movie3

Potential Rules:

Movie1 → Movie2

Movie2 → Movie4

Movie1 → Movie3

MARKET BASKET OPTIMISATION

Transaction ID	Products purchased
46578	Burgers, French Fries, Vegetables
98989	Burgers, French Fries, Ketchup
71527	Vegetables, Fruits
78981	Pasta, Fruits, Butter, Vegetables
89192	Burgers, Pasta, French Fries
61557	Fruits, Orange Juice, Vegetables
87923	Burgers, French Fries, Ketchup, Mayo

Potential Rules:

Burgers ⇒ French Fries

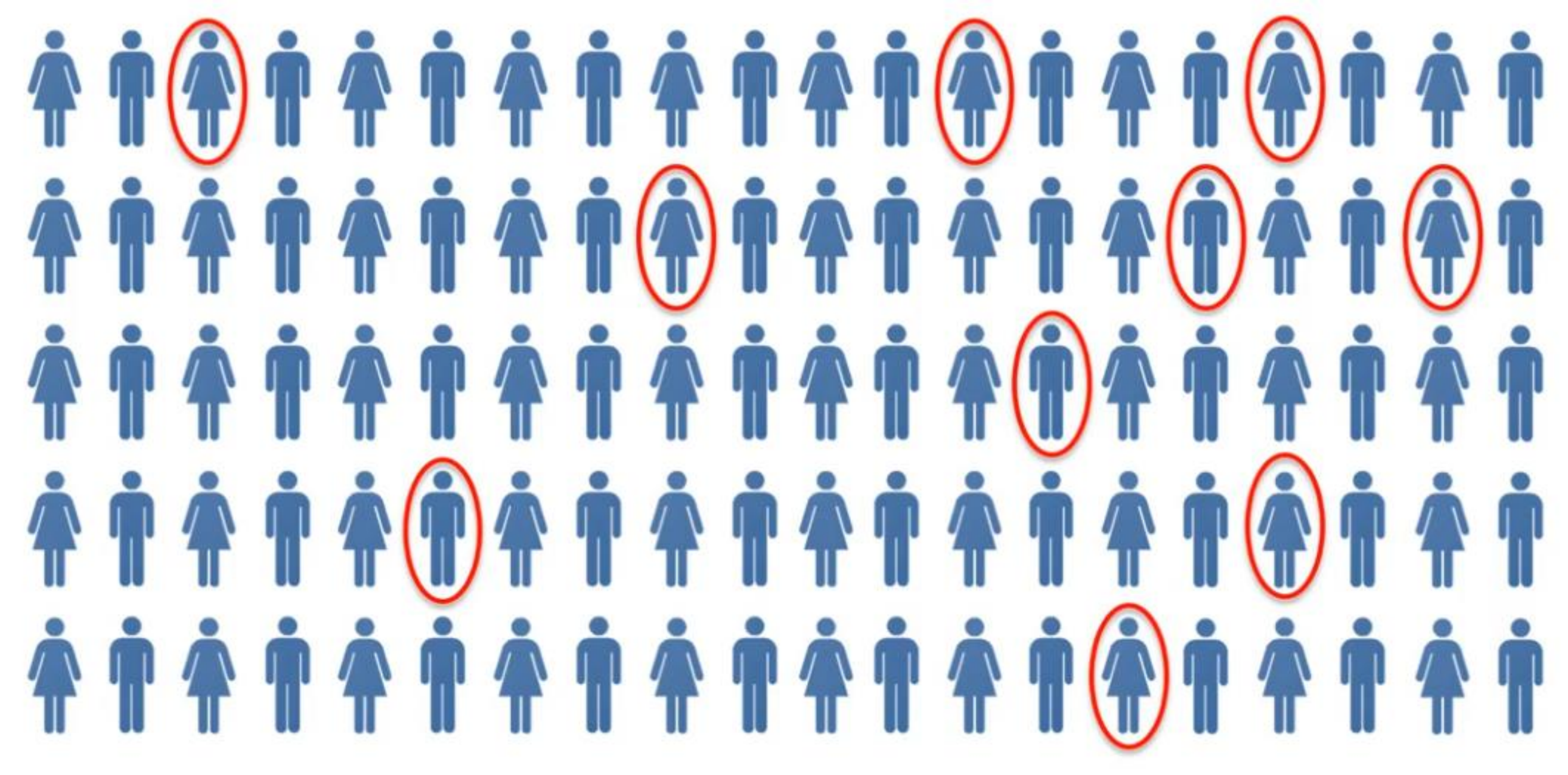
Vegetables ⇒ Fruits

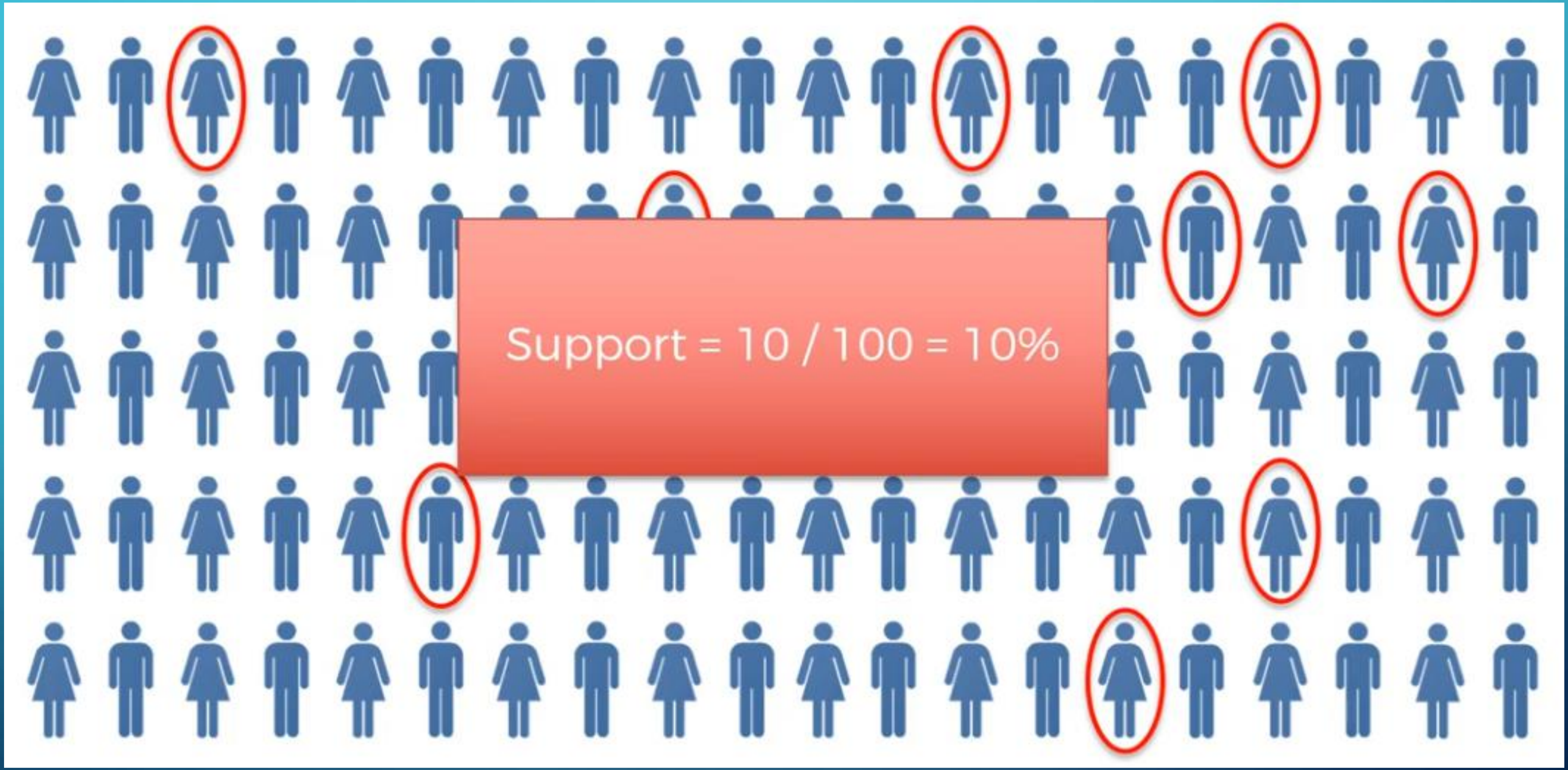
Burgers, French Fries ⇒ Ketchup

SUPPORT

Movie Recommendation: $$\text{support}(\boldsymbol{M}) = \frac{\#\ \text{user watchlists containing } \boldsymbol{M}}{\#\ \text{user watchlists}}$$

Market Basket Optimisation: $$\text{support}(\boldsymbol{I}) = \frac{\#\ \text{transactions containing } \boldsymbol{I}}{\#\ \text{transactions}}$$

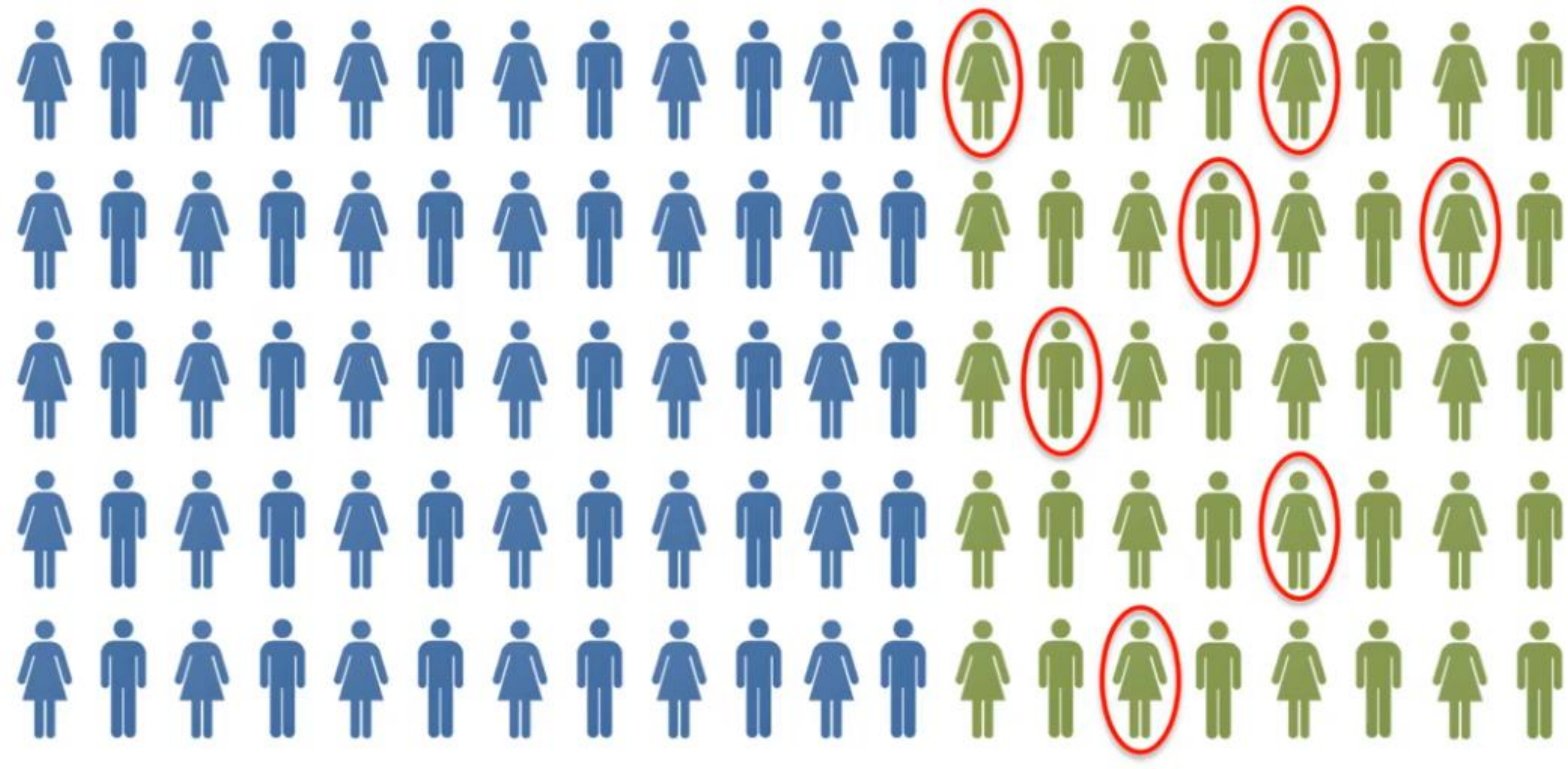


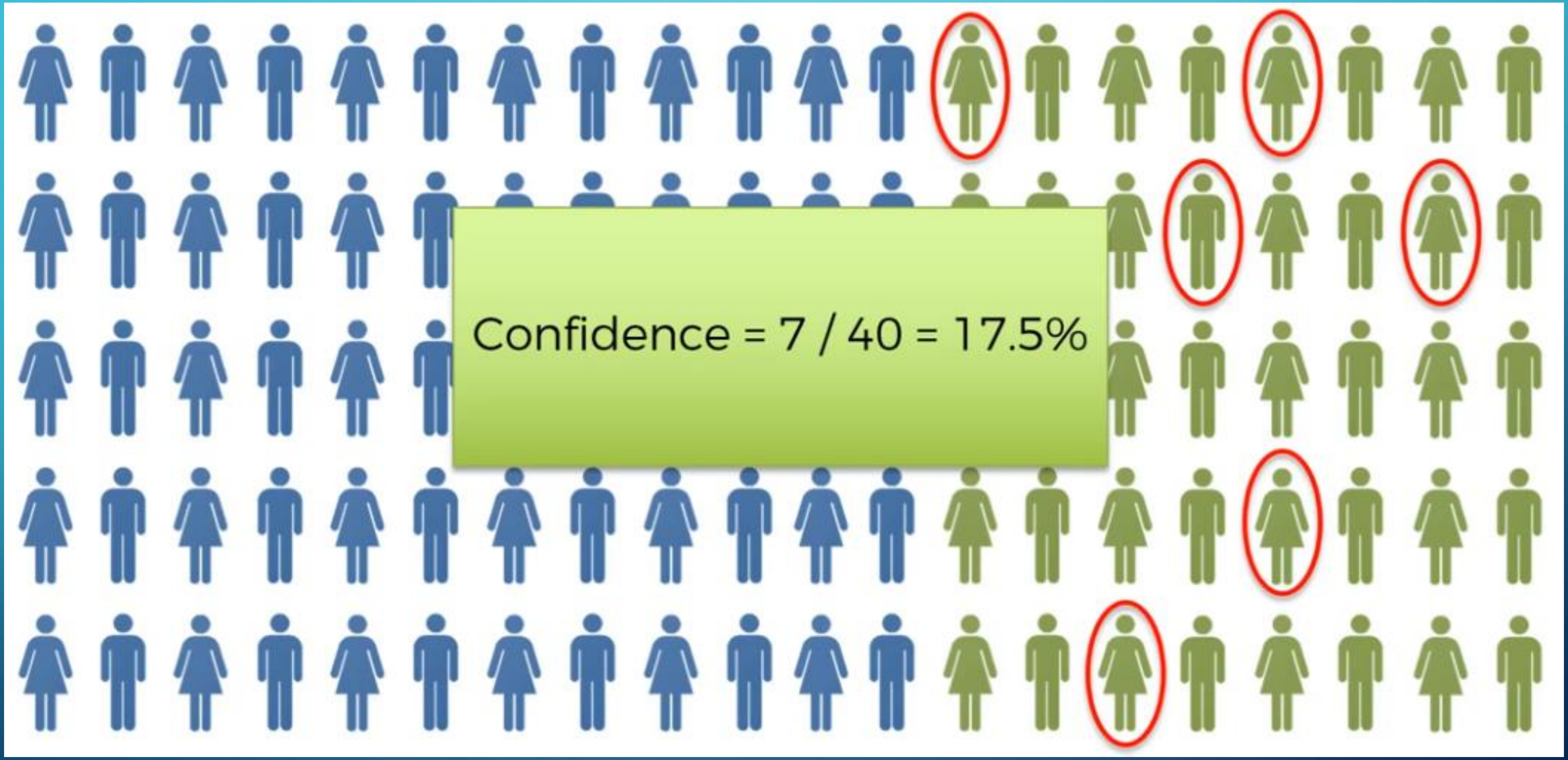

Support = 10 / 100 = 10%

CONDIFENCE

Movie Recommendation: $$\text{confidence}(M_1 \rightarrow M_2) = \frac{\#\ \text{user watchlists containing } M_1 \text{ and } M_2}{\#\ \text{user watchlists containing } M_1}$$

Market Basket Optimisation: $$\text{confidence}(I_1 \rightarrow I_2) = \frac{\#\ \text{transactions containing } I_1 \text{ and } I_2}{\#\ \text{transactions containing } I_1}$$

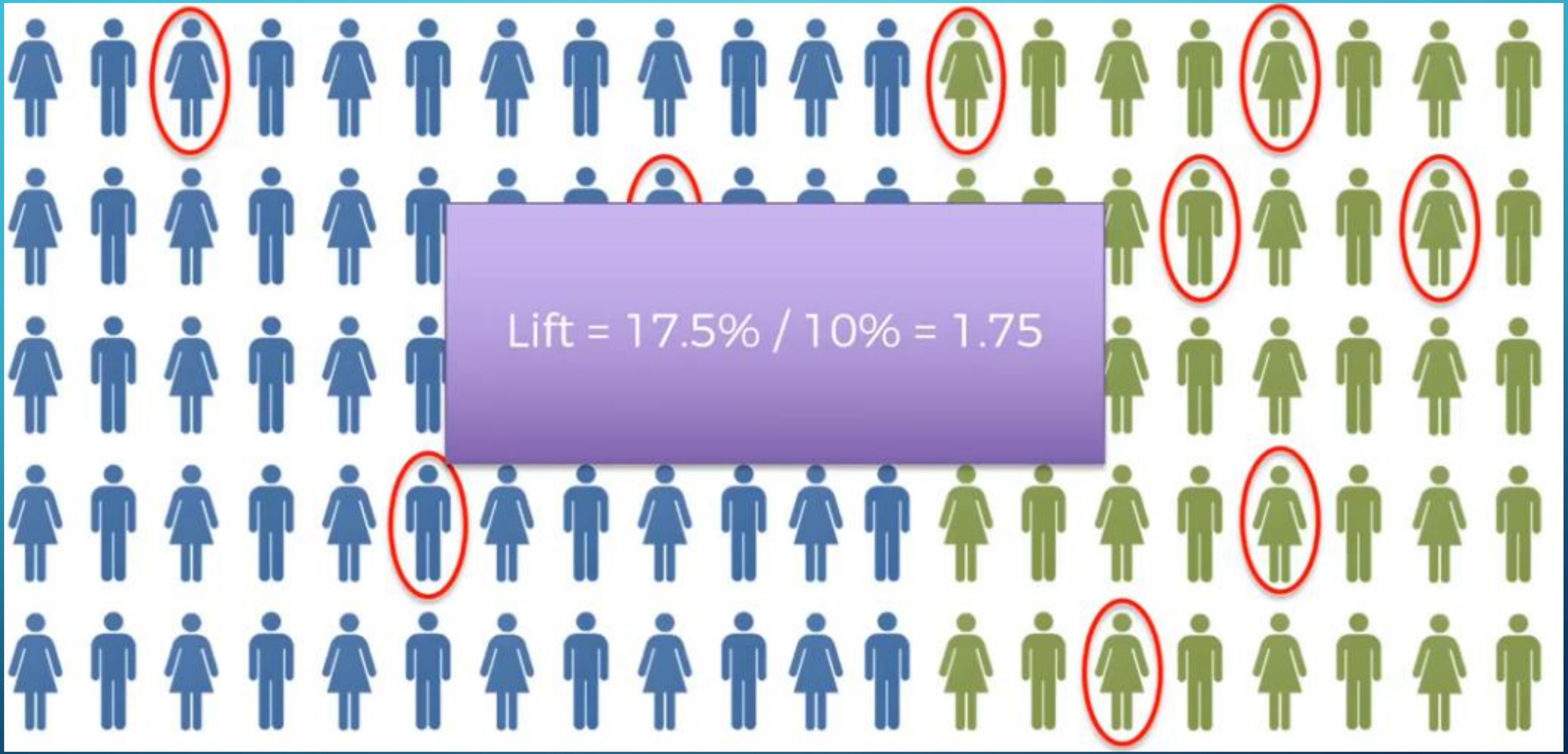



Confidence = 7 / 40 = 17.5%

LIFT

Movie Recommendation: $$\text{lift}(\boldsymbol{M_1} \rightarrow \boldsymbol{M_2}) = \frac{\text{confidence}(\boldsymbol{M_1} \rightarrow \boldsymbol{M_2})}{\text{support}(\boldsymbol{M_2})}$$

Market Basket Optimisation: $$\text{lift}(\boldsymbol{I_1} \rightarrow \boldsymbol{I_2}) = \frac{\text{confidence}(\boldsymbol{I_1} \rightarrow \boldsymbol{I_2})}{\text{support}(\boldsymbol{I_2})}$$


Lift = 17.5% / 10% = 1.75

STEPS

Step 1: Set a minimum support and confidence

↓

Step 2: Take all the subsets in transactions having higher support than minimum support

↓

Step 3: Take all the rules of these subsets having higher confidence than minimum confidence

↓

Step 4: Sort the rules by decreasing lift

TIME TO CODE!!