CLUSTERING

UNIVERSITY OF CALGARY

FOR LATER

DOWNLOAD THE

“Exercise - Clustering.zip”

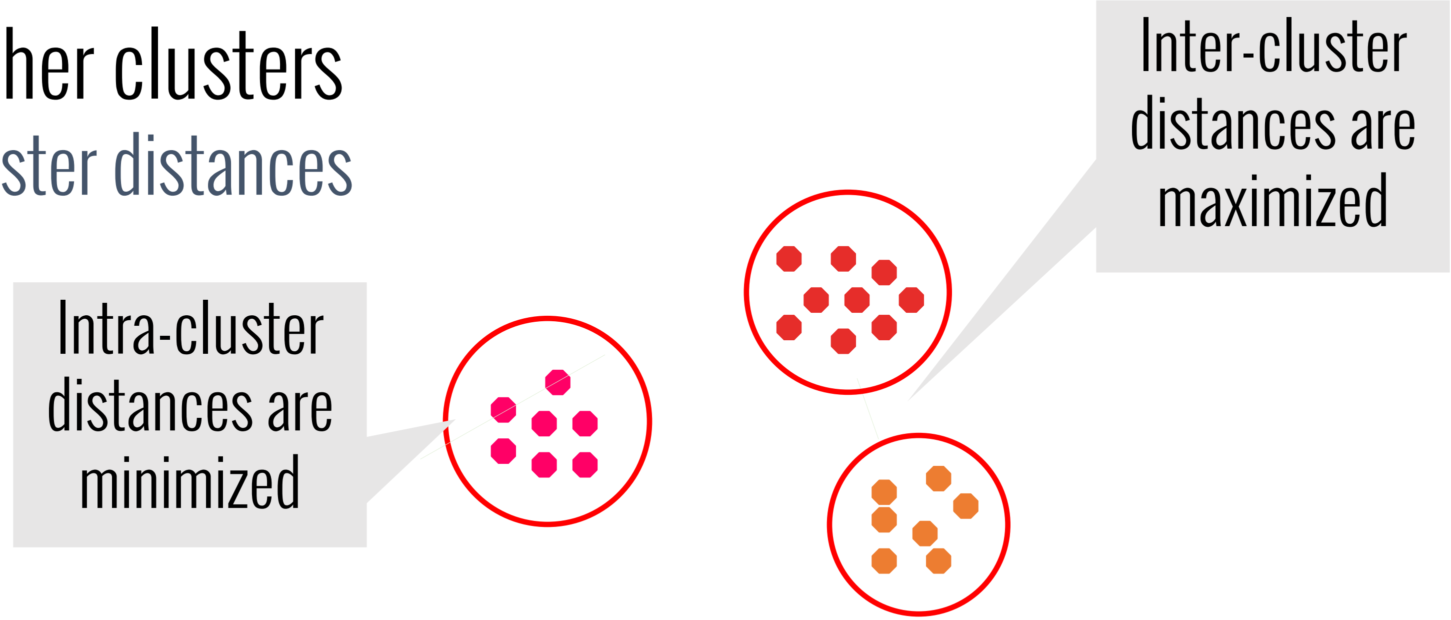
NOTEBOOK & DATASET

FROM THE COURSE SITE

CLUSTER ANALYSIS FINDS ‘INTERESTING’ GROUPS OF OBJECTS BASED ON SIMILARITY

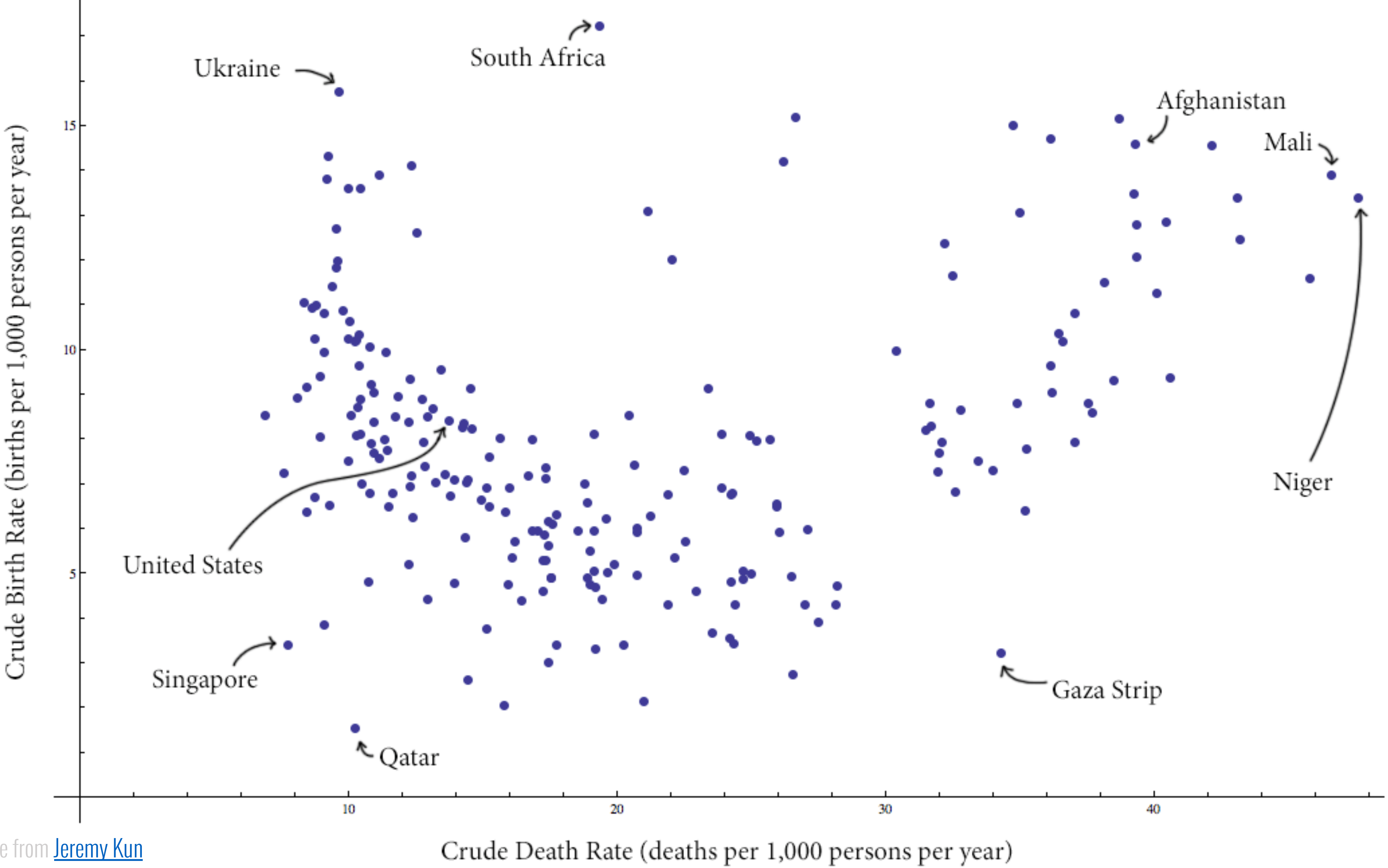
What typically makes a ‘good’ clustering?

- Members are highly similar to each other
  - Minimize within-cluster distances
- Well-separated from other clusters
  - Maximize between-cluster distances

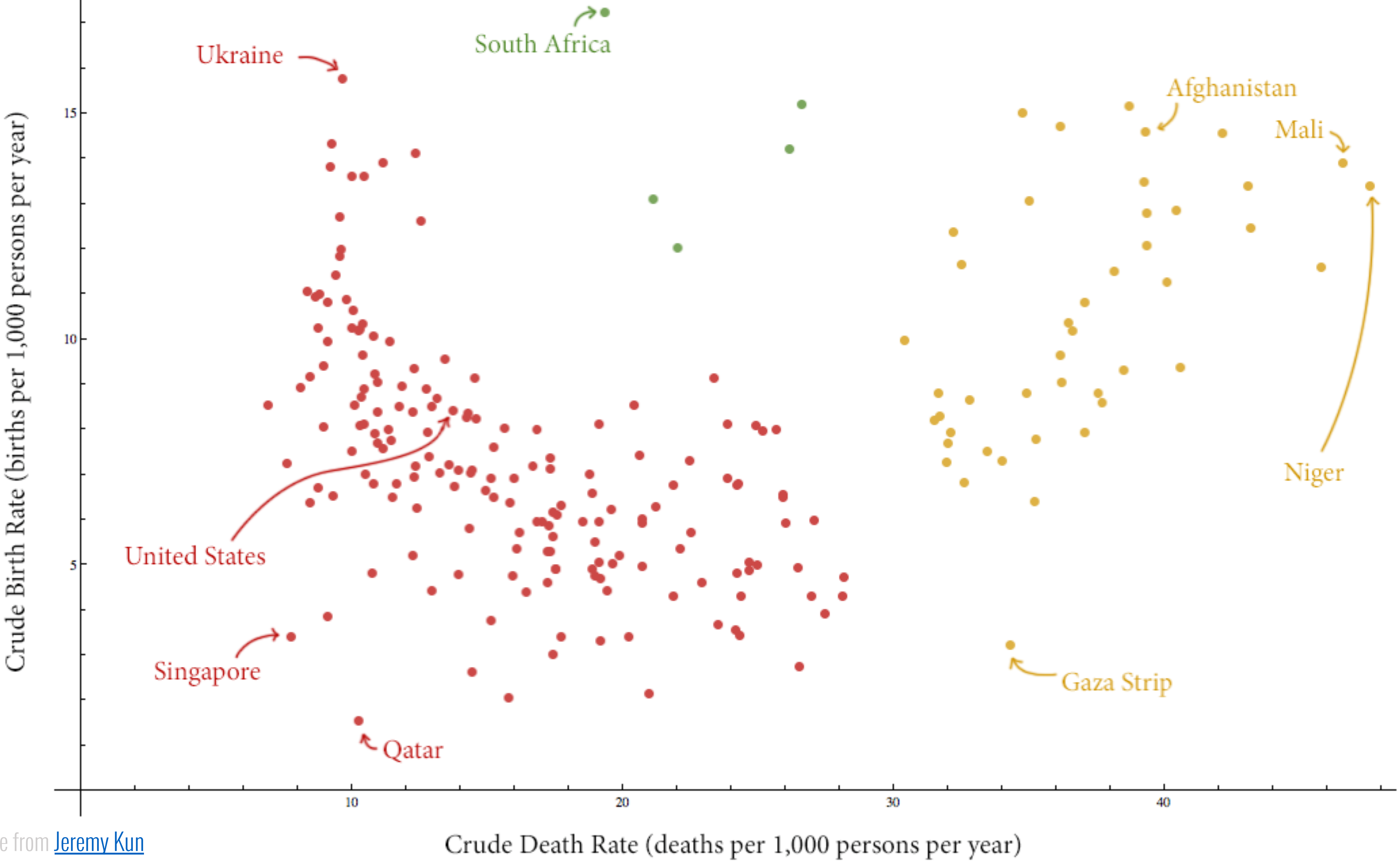


Inter-cluster distances are maximized

Intra-cluster distances are minimized



Example from Jeremy Kun



Example from Jeremy Kun

APPLICATIONS OF CLUSTER ANALYSIS

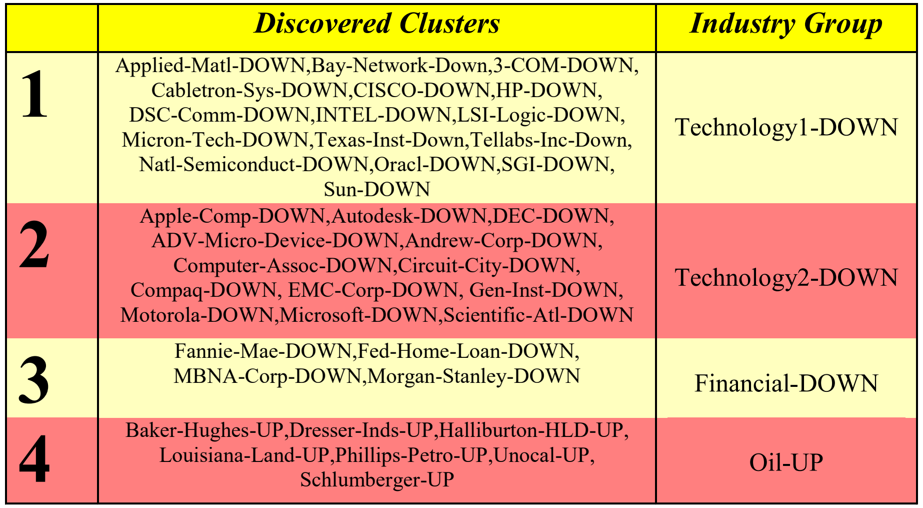
Understanding

Group related documents for browsing

Group genes and proteins with similar functionality

Group stocks with similar price fluctuations

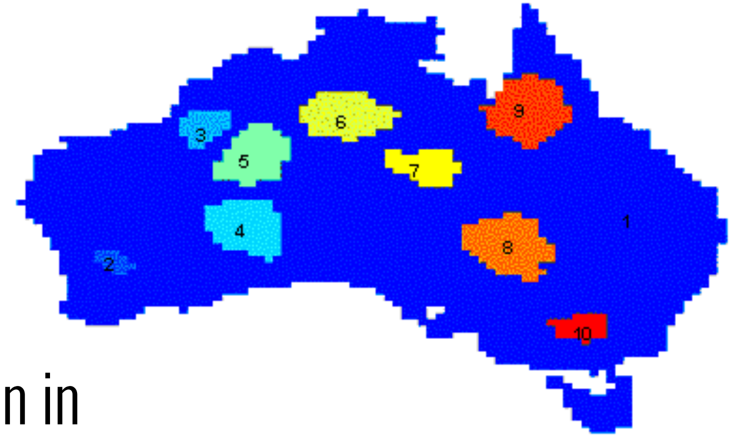
…



	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

Summarization

Reduce size of large data sets



Clustering precipitation in Australia

RELATIONSHIP TO OTHER APPROACHES

	continuous	categorical
supervised	**regression**	**classification**
unsupervised	dimension reduction	clustering

SUMMARY: CONDUCTING CLUSTER ANALYSIS

Formulate the Problem

Select a Distance/Similarity Measure

Select a Clustering Procedure

Decide on the Number of Clusters

Interpret and Profile Clusters

Assess the Validity of Clustering

CLUSTERING IS OFTEN USED AS AN EXPLORATORY DATA ANALYSIS TOOL

Data understanding

- Finding underlying factors, groups, structure

Data navigation

- Creating hierarchies to support browsing

Data reduction

- Clustering creates a new nominal variable that can be used in any further analysis.
- A good way to quantize variable measures into non-uniform buckets

Data cleaning / smoothing

- Infer or interpolate missing attributes from cluster neighbors

CLUSTERING ARISES NATURALLY IN MANY FIELDS

- Business
  - Market segments
  - Web site visitors
- Social network analysis
  - Find communities
- Information Retrieval
  - Search results clustered by similarity, event or topic
  - Personalization for groups of similar users
- Health
  - DNA gene expression
    - Cluster cancer variants into treatment groups, based on immunomarkers of cell samples
  - Medical imaging
    - Find likely tumors

SUMMARY: CONDUCTING CLUSTER ANALYSIS

Formulate the Problem

Select a Distance/Similarity Measure

Select a Clustering Procedure

Decide on the Number of Clusters

Interpret and Profile Clusters

Assess the Validity of Clustering

SELECTING A DISTANCE METRIC

Your distance function determines the clusters you'll get

Can be formulated in different (opposite) ways.

- Maximize *intra*-cluster similarity while minimizing *inter*-cluster similarity
- Minimize *intra*-cluster distances while maximizing *inter*-cluster distances

But what's the right metric for distance/similarity?

- How do we calculate genetic similarity?
- How do we calculate the similarity in musical taste?
- How do we calculate the similarity in car features?

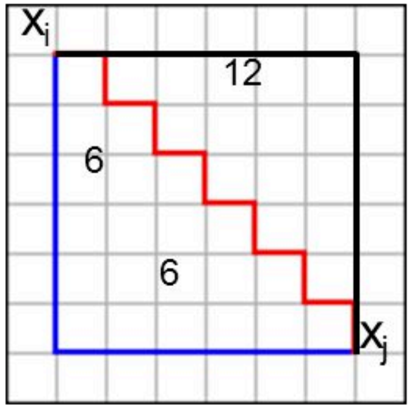
DATAFRAME TO SIMILARITY MATRIX

Entry #	Variable 1	Variable 2	Variable 3
1	A	True	3.5
2	A	False	4.5
3	B	True	5.6

⬇

	1	2	3
1	Sim(1,1)	Sim(1,2)	Sim(1,3)
2	Sim(2,1)	Sim(2,2)	Sim(2,3)
3	Sim(3,1)	Sim(3,2)	Sim(3,3)

SOME OPTIONS

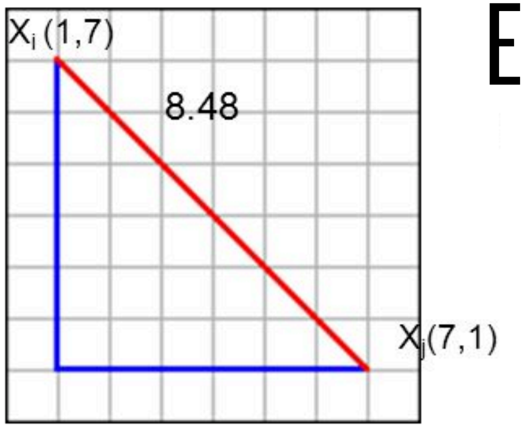


Manhattan distance

Think of it as city grids

$|x_1-x_2| + |y_1-y_2| + |z_1-z_2| + \ldots$

Not often used



Euclidean distance

$\mathrm{sqrt}((x_1-x_2)^2 + (y_1-y_2)^2 + (z_1-z_2)^2 + \ldots)$

Good for: quantitative data, equal importance/weights, normalized data

SOME OPTIONS

Jaccard distance

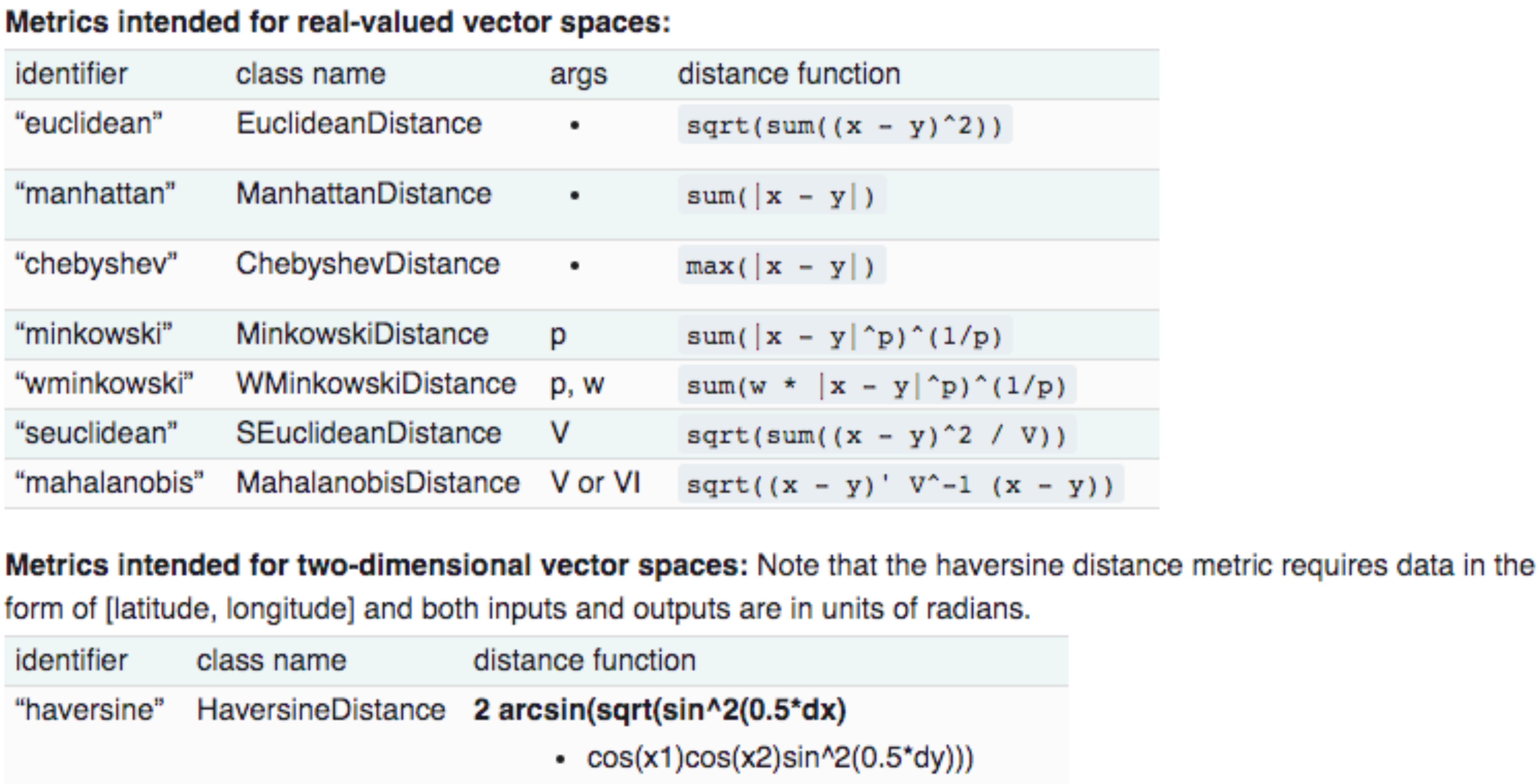
When data is nominal (true/false, words, etc.)

$$\frac{|intersection(A,B)|}{|union(A,B)|}$$

“How many things we have in total versus how many things we have in common”

YOU HAVE LOTS OF CHOICES

For example, SciKit-Learn supports:

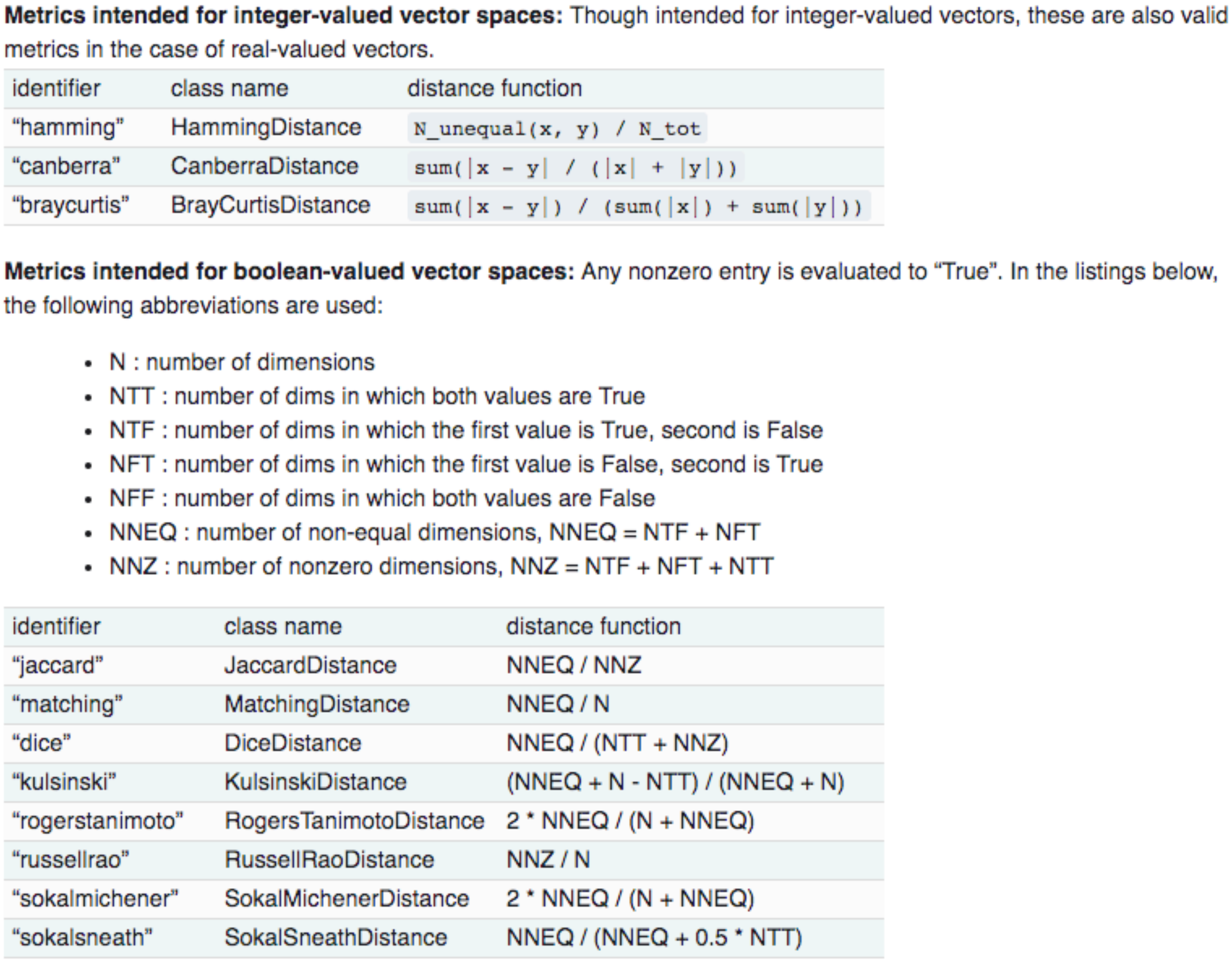


Metrics intended for real-valued vector spaces:

identifier	class name	args	distance function
“euclidean”	EuclideanDistance	•	`sqrt(sum((x - y)^2))`
“manhattan”	ManhattanDistance	•	`sum(|x - y|)`
“chebyshev”	ChebyshevDistance	•	`max(|x - y|)`
“minkowski”	MinkowskiDistance	p	`sum(|x - y|^p)^(1/p)`
“wminkowski”	WMinkowskiDistance	p, w	`sum(w * |x - y|^p)^(1/p)`
“seuclidean”	SEuclideanDistance	V	`sqrt(sum((x - y)^2 / V))`
“mahalanobis”	MahalanobisDistance	V or VI	`sqrt((x - y)' V^-1 (x - y))`

Metrics intended for two-dimensional vector spaces: Note that the haversine distance metric requires data in the form of [latitude, longitude] and both inputs and outputs are in units of radians.

identifier	class name	distance function
“haversine”	HaversineDistance	**2 arcsin(sqrt(sin^2(0.5*dx)** • cos(x1)cos(x2)sin^2(0.5*dy)))



Metrics intended for integer-valued vector spaces: Though intended for integer-valued vectors, these are also valid metrics in the case of real-valued vectors.

identifier	class name	distance function
"hamming"	HammingDistance	`N_unequal(x, y) / N_tot`
"canberra"	CanberraDistance	`sum(|x - y| / (|x| + |y|))`
"braycurtis"	BrayCurtisDistance	`sum(|x - y|) / (sum(|x|) + sum(|y|))`

Metrics intended for boolean-valued vector spaces: Any nonzero entry is evaluated to "True". In the listings below, the following abbreviations are used:

- N : number of dimensions
- NTT : number of dims in which both values are True
- NTF : number of dims in which the first value is True, second is False
- NFT : number of dims in which the first value is False, second is True
- NFF : number of dims in which both values are False
- NNEQ : number of non-equal dimensions, NNEQ = NTF + NFT
- NNZ : number of nonzero dimensions, NNZ = NTF + NFT + NTT

identifier	class name	distance function
"jaccard"	JaccardDistance	NNEQ / NNZ
"matching"	MatchingDistance	NNEQ / N
"dice"	DiceDistance	NNEQ / (NTT + NNZ)
"kulsinski"	KulsinskiDistance	(NNEQ + N - NTT) / (NNEQ + N)
"rogerstanimoto"	RogersTanimotoDistance	2 * NNEQ / (N + NNEQ)
"russellrao"	RussellRaoDistance	NNZ / N
"sokalmichener"	SokalMichenerDistance	2 * NNEQ / (N + NNEQ)
"sokalsneath"	SokalSneathDistance	NNEQ / (NNEQ + 0.5 * NTT)

SIMILARITY METRIC CHOICES

Try to use metrics that you **understand** and **can reason about**!

Very often you'll **roll your own** or use a known similarity/distance metric for your specific type of data.

Need to be careful if the metric is

similarity (0 is far, 1 is close)
or
dissimilarity (1 is far, 0 is close)

SUMMARY: CONDUCTING CLUSTER ANALYSIS

Formulate the Problem

Select a Distance/Similarity Measure

Select a Clustering Procedure

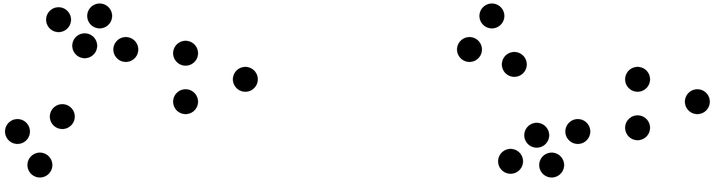
Decide on the Number of Clusters

Interpret and Profile Clusters

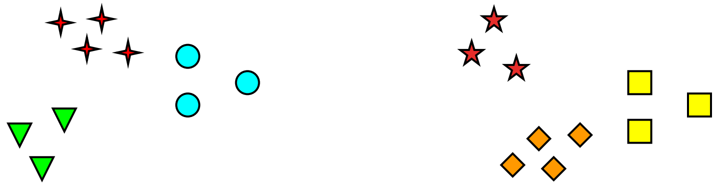
Assess the Validity of Clustering

SELECTING A CLUSTERING TECHNIQUE

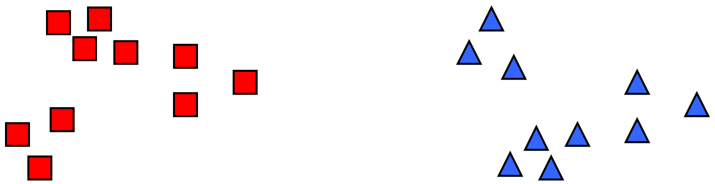
CLUSTERING CAN BE AMBIGUOUS:
WHAT IS THE ‘BEST’ CLUSTERING HERE?



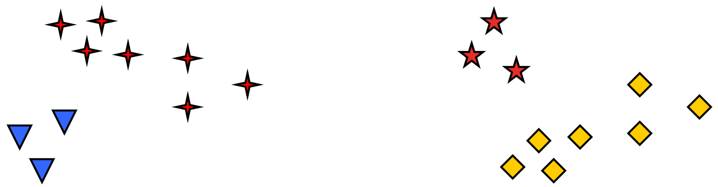
How many clusters?



Six Clusters



Two Clusters



Four Clusters

THERE ARE LOTS OF CLUSTERING APPROACHES

Assigning objects to clusters

- 'Hard' (partitional) each object belongs to exactly 1 cluster
- 'Soft' : each object can belong to multiple clusters

Hierarchical vs non-hierarchical

- A set of nested clusters organized as a tree

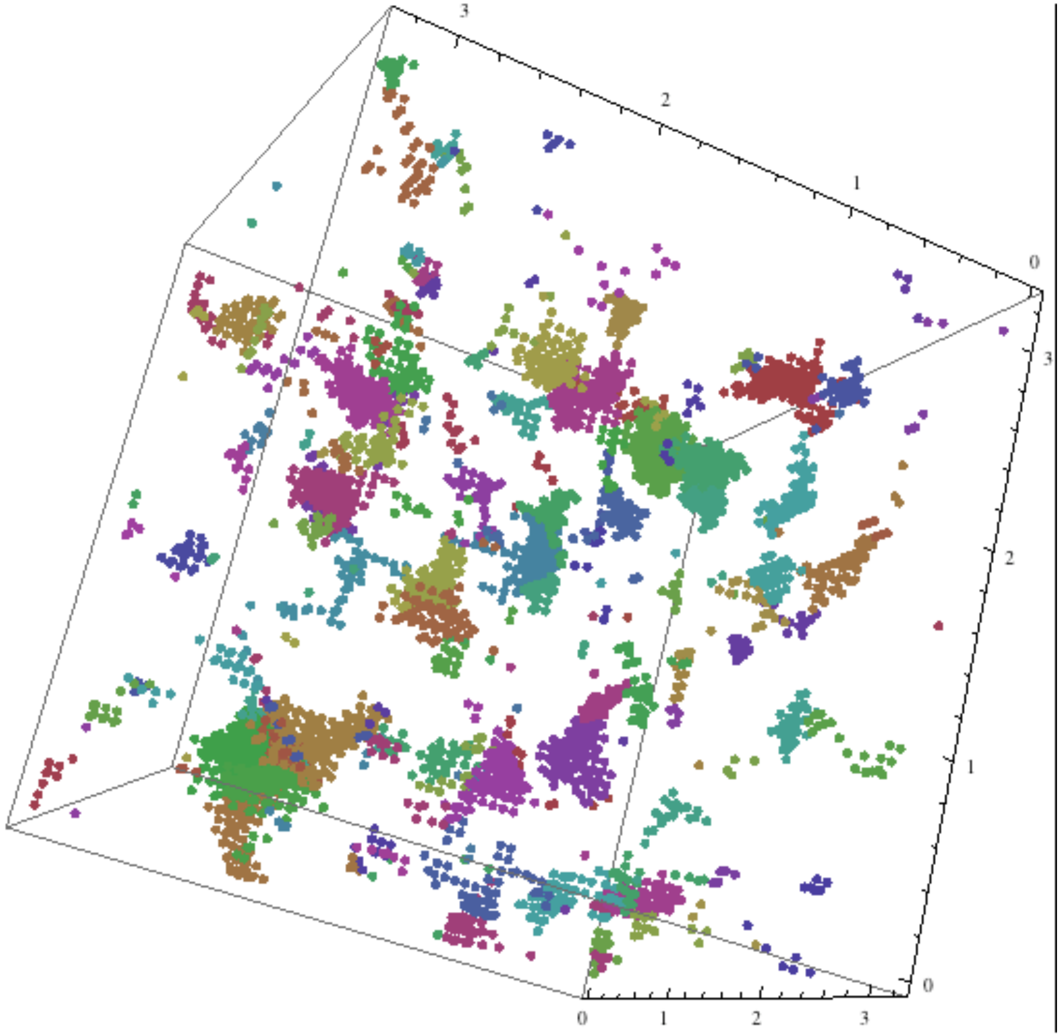
By far most widely-used fall into two types:

- **Hierarchical**: agglomerative, single-link, etc.
- **Partitional**: k-means, k-median, etc.

CLUSTERING AND DIMENSIONALITY

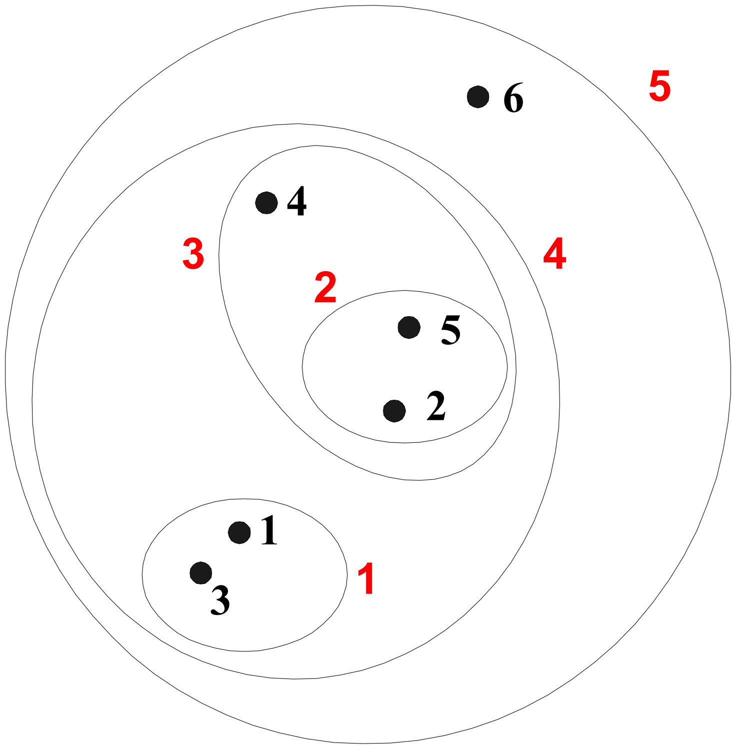
Most of our examples will be 2D (its easy to illustrate).

But remember that your distance/similarity metrics can include **as many dimensions as you want!**

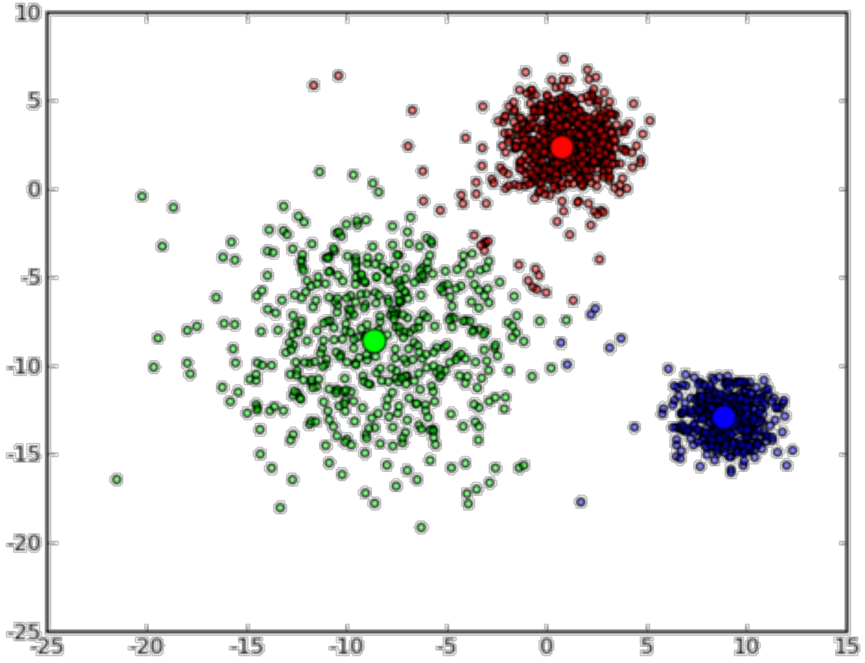


TODAY

HIERARCHICAL CLUSTERING



K-MEANS CLUSTERING

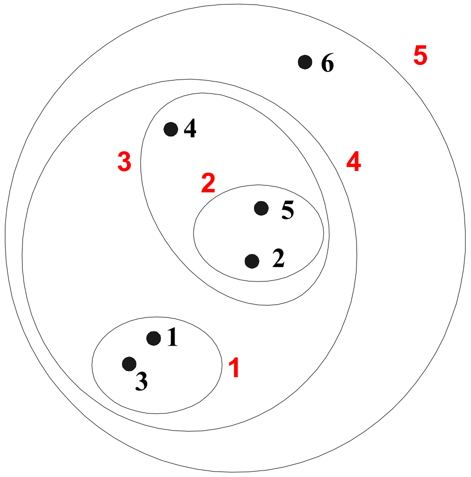


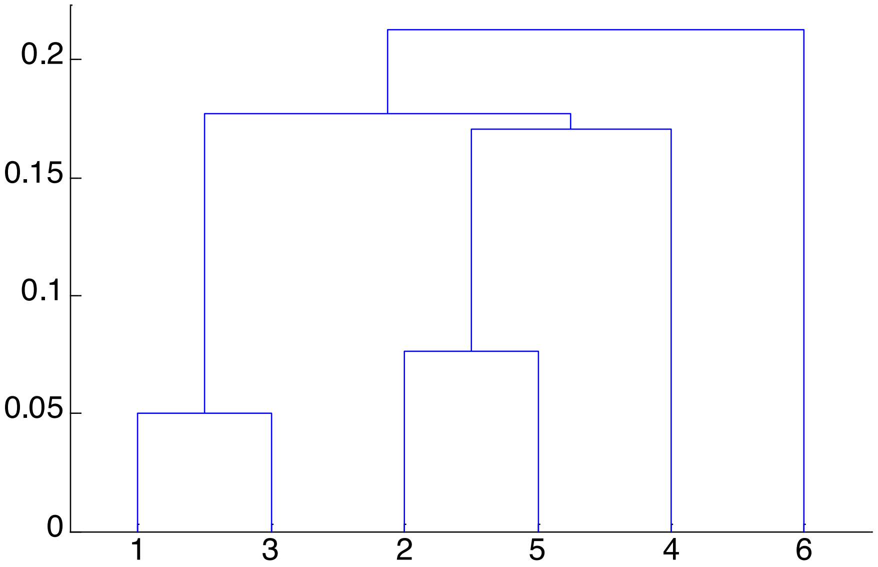
HIERARCHICAL CLUSTERING

Produces a set of nested clusters organized as a **hierarchical tree**

Can be visualized as a **dendrogram**

A tree like diagram that records the sequences of merges or splits





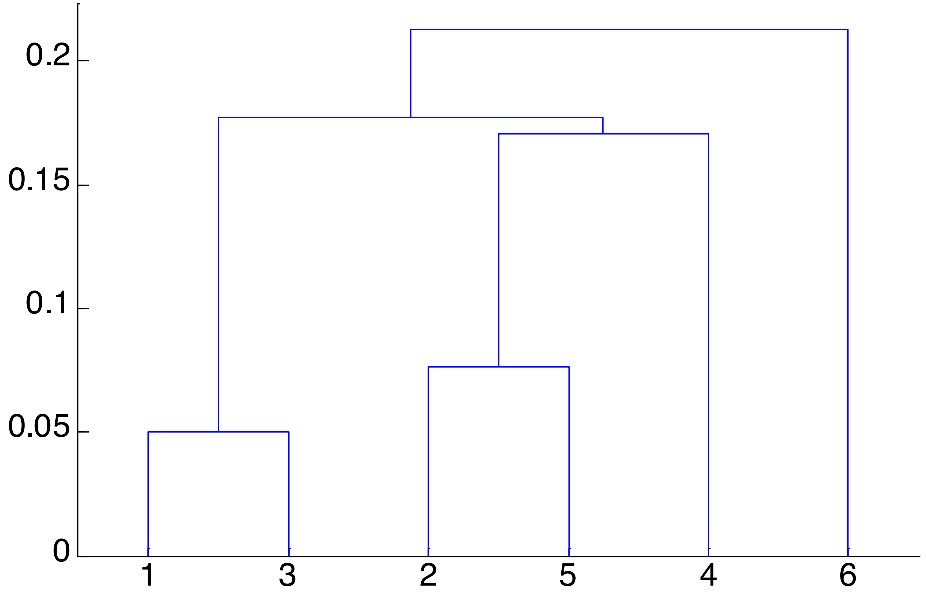
STRENGTHS OF HIERARCHICAL CLUSTERING

Do not have to assume any particular number of clusters

- Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level

They may correspond to meaningful taxonomies

- Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)



HIERARCHICAL CLUSTERING

Bottom-up ('Agglomerative')

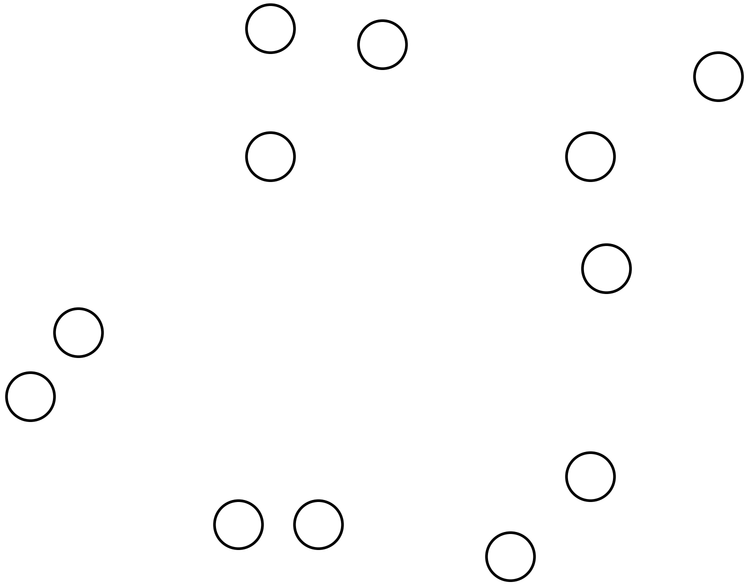
- Start with each point being in its own cluster
- At each step
  - Merge the most similar pair of clusters based on a cost function
  - Continue until you have k clusters, or everything is in one big cluster

Top-down ('Divisive')

- Start with all points in a single big cluster
- At each step:
  - Split the cluster into two smaller clusters based on a cost function
  - Continue until you have k clusters, or each point is in its own cluster

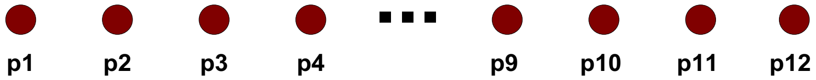
AGGLOMERATIVE (BOTTOM-UP) CLUSTERING: STARTING SITUATION

Start with clusters of individual points and a proximity matrix of object-to-object distances



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Similarity Matrix



AGGLOMERATIVE CLUSTERING ALGORITHM

One popular hierarchical clustering technique

For this discussion
“proximity”=“similarity”=“distance”

Basic algorithm is straightforward

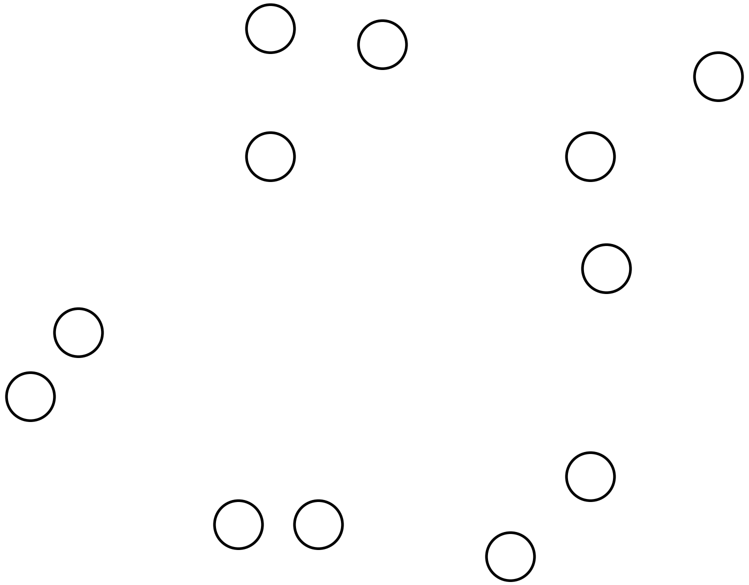
```
1.  Compute the proximity matrix
2.  Let each data point be a cluster
3.  Repeat:
4.          Merge the two closest clusters
5.          Update the proximity matrix
6.          Stop if only a single cluster remains
```

Key operation: computation of the proximity of two clusters. The <u>cost function</u>.

Different approaches to defining the distance between clusters distinguish the different algorithms

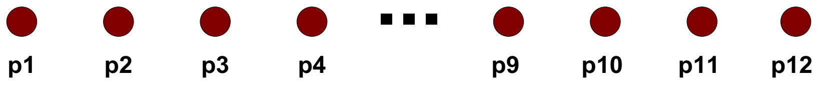
AGGLOMERATIVE (BOTTOM-UP) CLUSTERING

Start with clusters of individual points and a similarity matrix of object-to-object distances



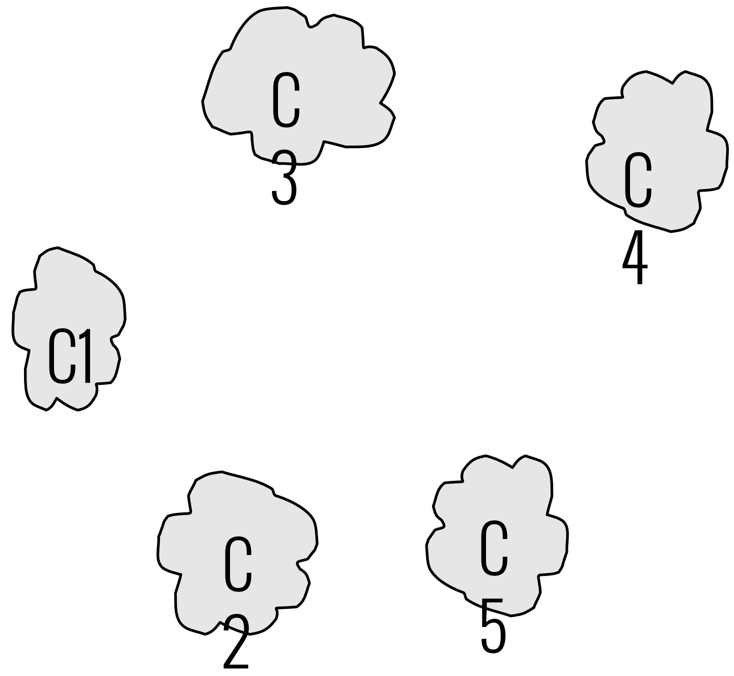
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Similarity Matrix



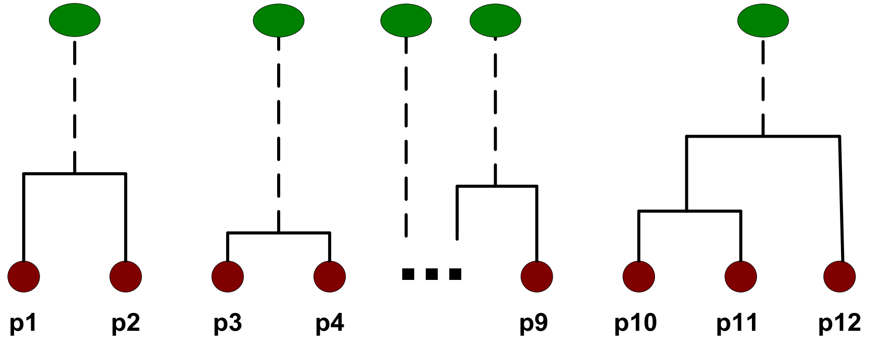
INTERMEDIATE SITUATION

After some merging steps, we have some clusters



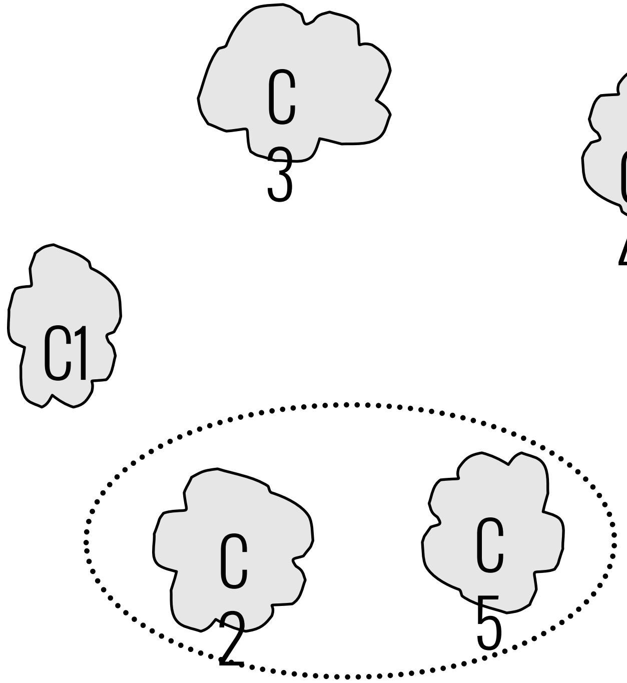
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Similarity Matrix



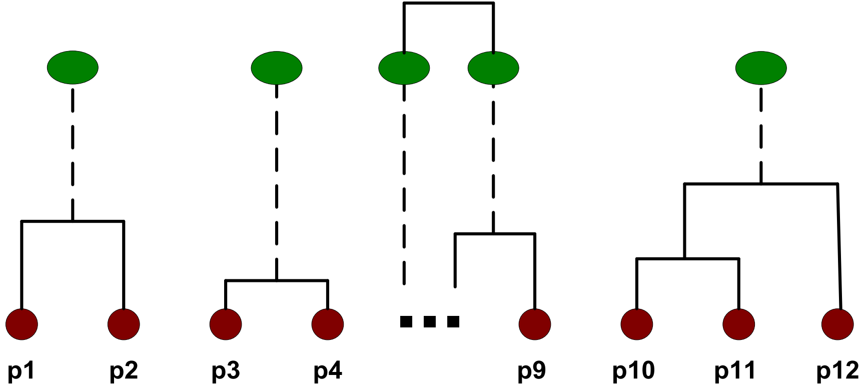
INTERMEDIATE SITUATION

We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



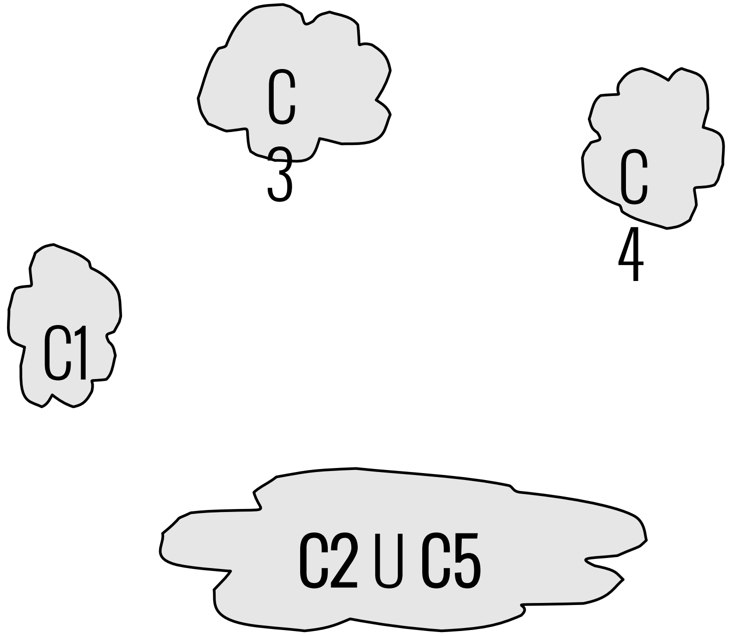
	C1	C2	C3	C4	C5
C1		▨			▨
C2	▨	▨	▨	▨	▨
C3		▨			▨
C4		▨			▨
C5	▨	▨	▨	▨	▨

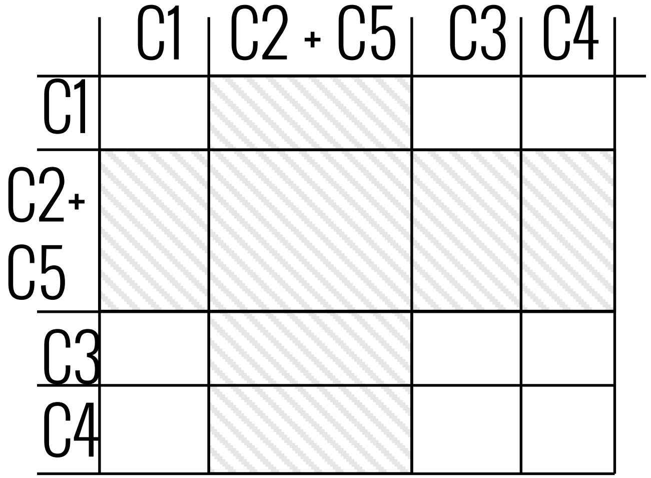
Similarity Matrix



AFTER MERGING

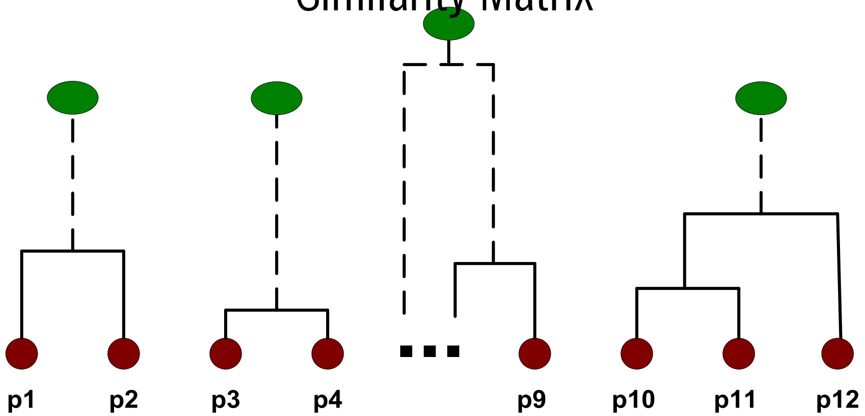
The question is “How do we update the proximity matrix?”



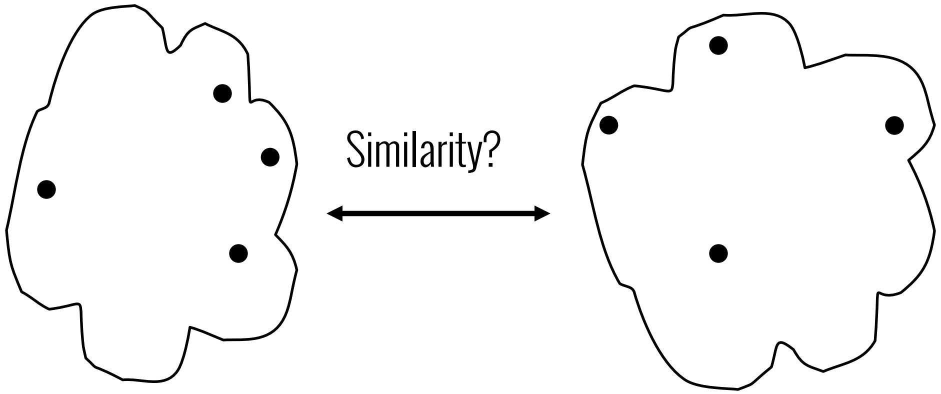


	C1	C2 + C5	C3	C4
C1		▨		
C2+ C5	▨	▨	▨	▨
C3		▨		
C4		▨		

Similarity Matrix



HOW TO DEFINE INTER-CLUSTER SIMILARITY



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Similarity Matrix

MIN

MAX

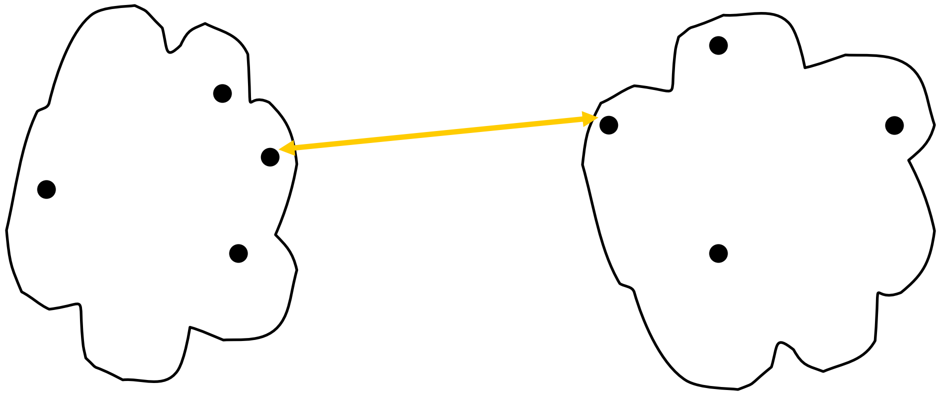
Group Average

Distance Between Centroids

Other methods driven by an objective function

Ward’s Method uses squared error

HOW TO DEFINE INTER-CLUSTER SIMILARITY



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Similarity Matrix

MIN

MAX

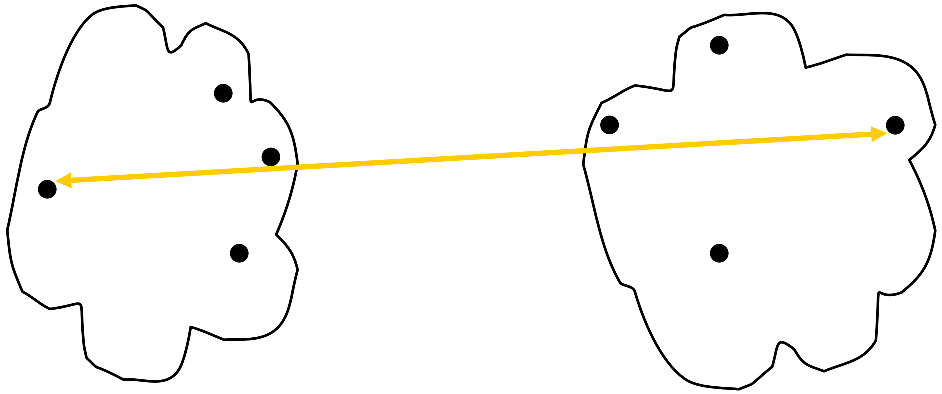
Group Average

Distance Between Centroids

Other methods driven by an objective function

- Ward's Method uses squared error

HOW TO DEFINE INTER-CLUSTER SIMILARITY



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Similarity Matrix

MIN

MAX

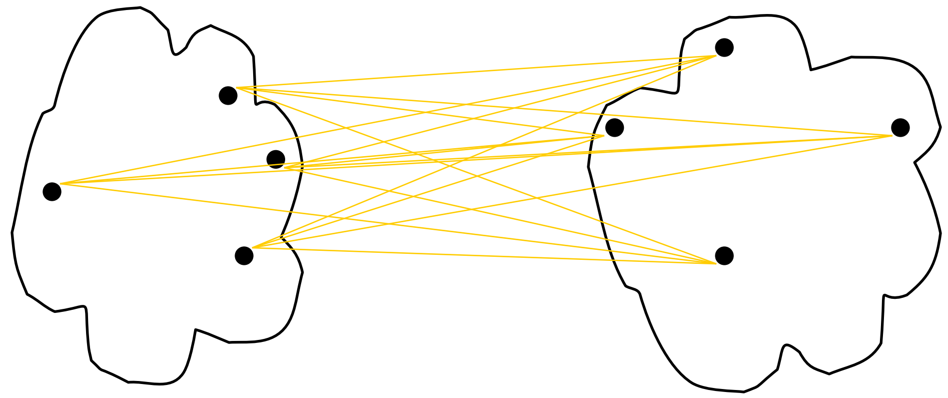
Group Average

Distance Between Centroids

Other methods driven by an objective function

- Ward's Method uses squared error

HOW TO DEFINE INTER-CLUSTER SIMILARITY



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Similarity Matrix

MIN

MAX

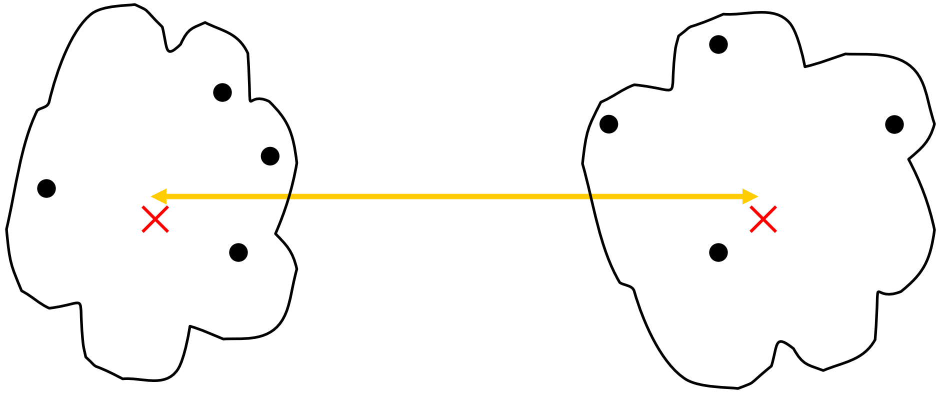
Group Average

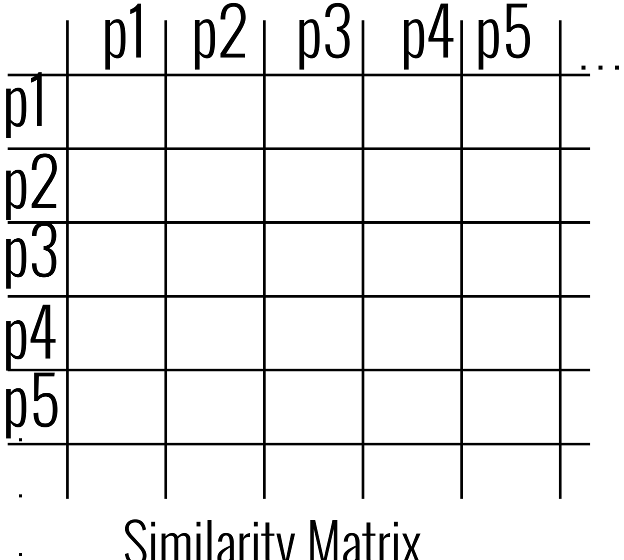
Distance Between Centroids

Other methods driven by an objective function

- Ward's Method uses squared error

HOW TO DEFINE INTER-CLUSTER SIMILARITY





Similarity Matrix

MIN

MAX

Group Average

Distance Between Centroids

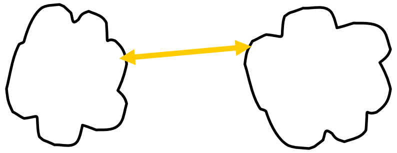
Other methods driven by an objective function

- Ward’s Method uses squared error

COST FUNCTIONS FOR BOTTOM-UP (AGGLOMERATIVE) CLUSTERING

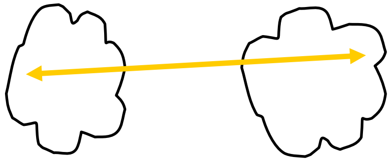
Single linkage

Minimum distance between clusters



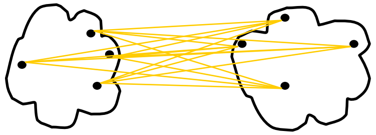
Complete linkage

Max distance between clusters



Average linkage

Average distance between clusters



WARD'S METHOD (1963)

Ward's distance between clusters C_i and C_j is the *difference* between the *total within cluster sum of squares for the two clusters separately*, and the *within cluster sum of squares resulting from merging the two clusters* in cluster C_{ij}

$$D_w(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

r_i: centroid of C_i
r_j: centroid of C_j
r_{ij}: centroid of C_{ij}

WARD'S DISTANCE FOR CLUSTERS

Similar to group average and centroid distance

Less susceptible to noise and outliers

Hierarchical analogue of k-means

- Can be used to initialize k-means

WHICH TYPE OF HIERARCHICAL CLUSTERING TO USE?

Different methods have different strengths and weaknesses:

Ward's method tends to give equal sized clusters
Single linkage (nearest neighbor) tends to make long strings into a cluster.

Top-down is sensitive to early errors
Bottom-up can’t see the whole dataset

SUMMARY: CONDUCTING CLUSTER ANALYSIS

Formulate the Problem

Select a Distance/Similarity Measure

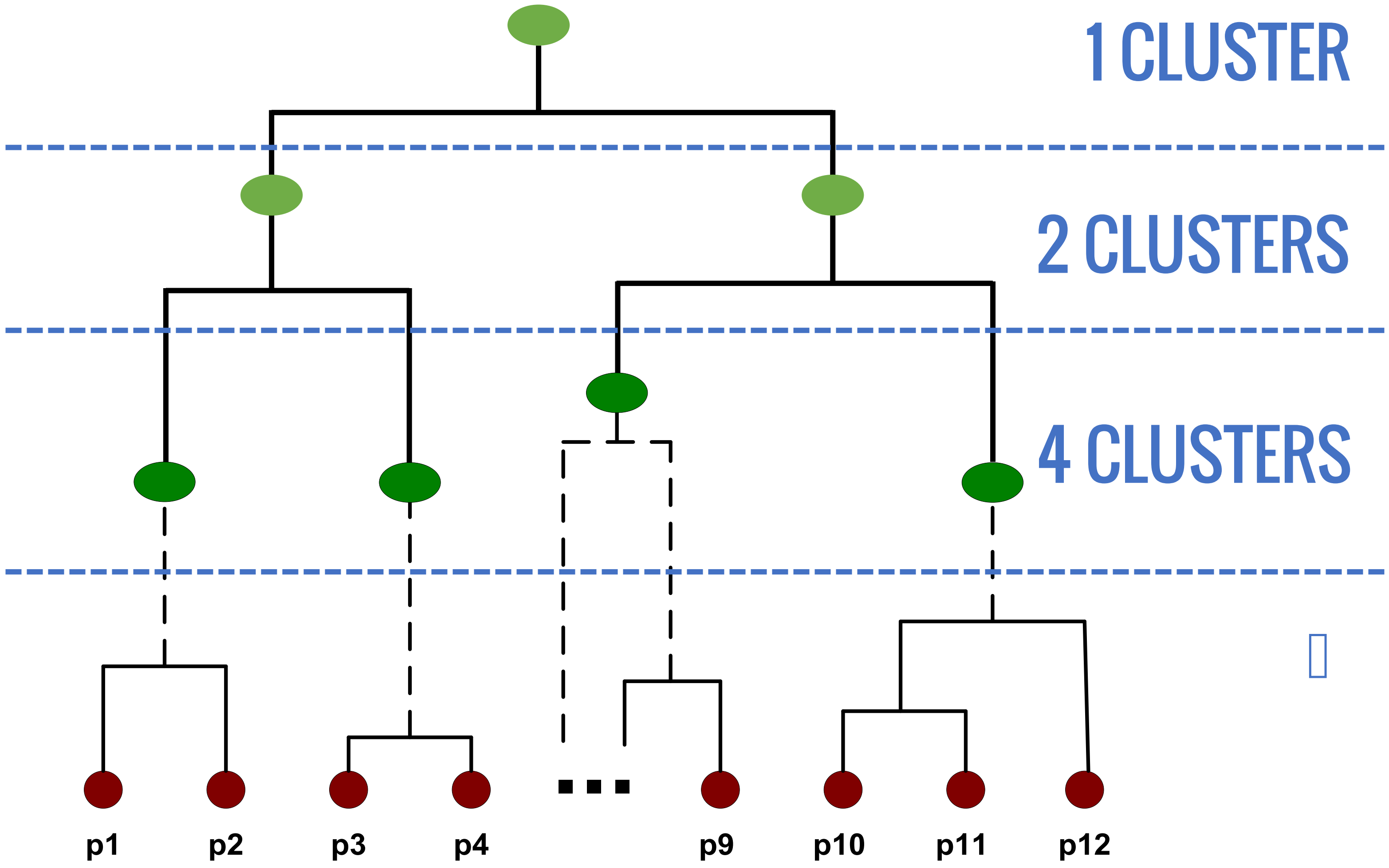
Select a Clustering Procedure

Decide on the Number of Clusters

Interpret and Profile Clusters

Assess the Validity of Clustering

CUTTING THE TREE INTO CLUSTERS

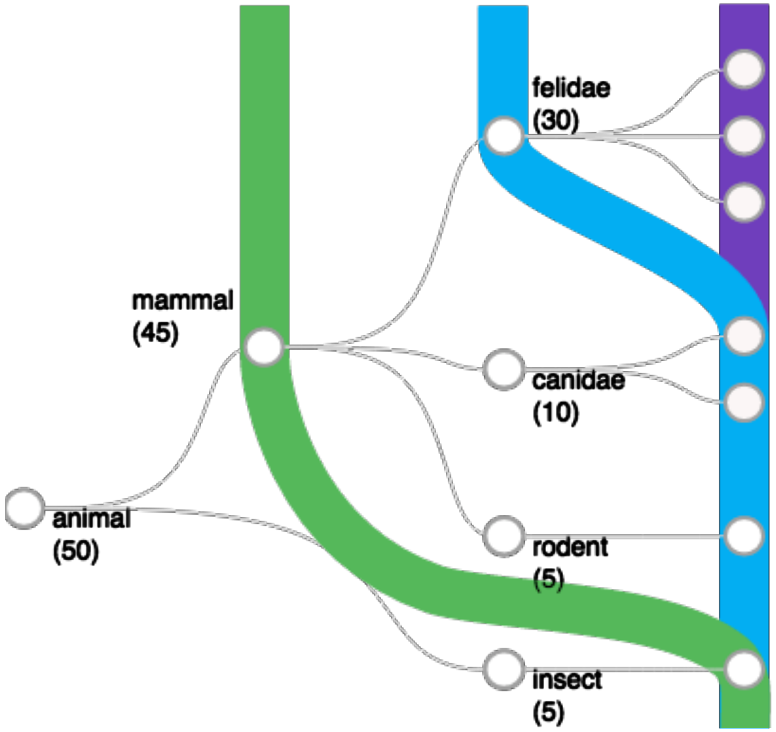


1 CLUSTER

2 CLUSTERS

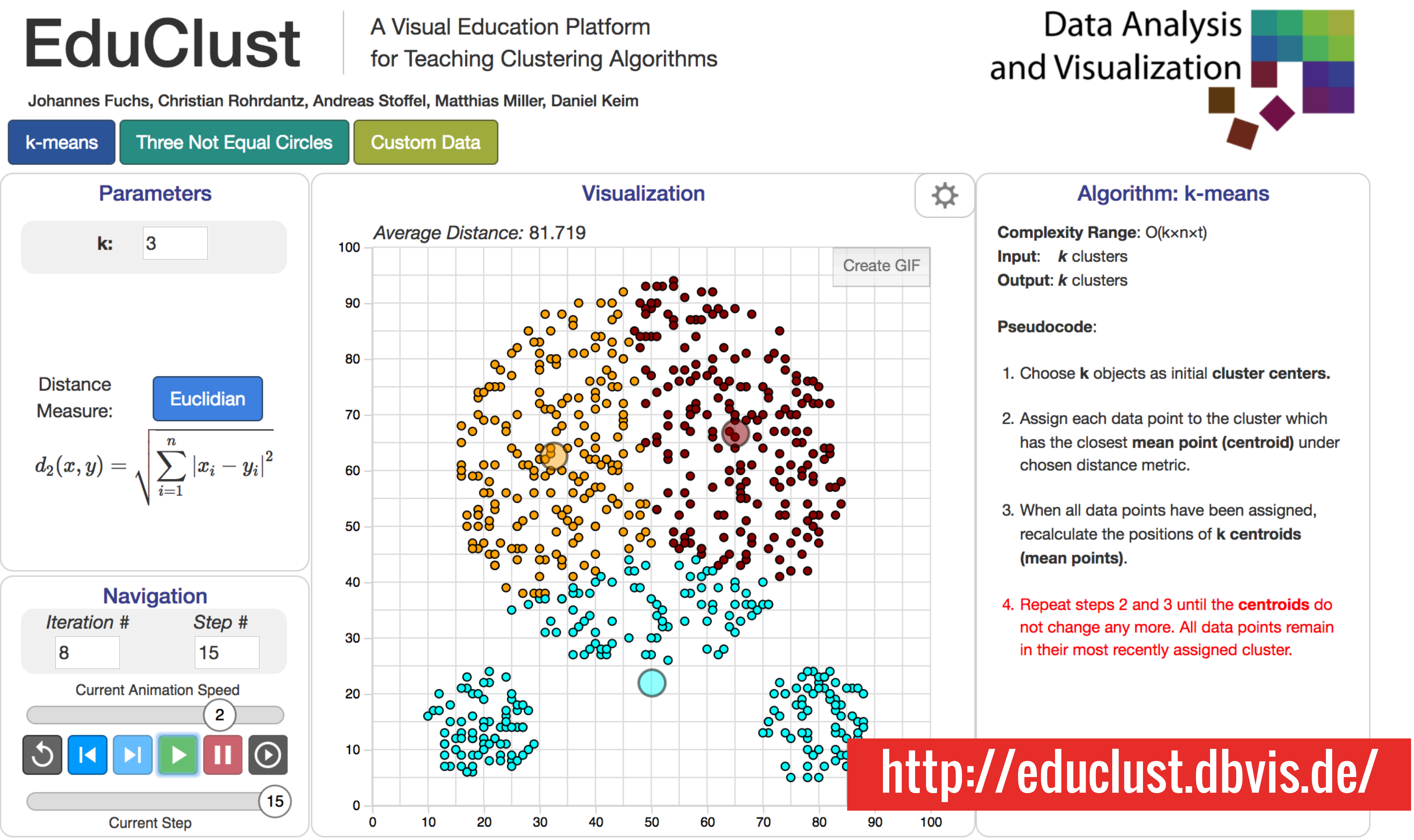
4 CLUSTERS

VARIABLE-DEPTH CUTS



	Value	Estimate	Error
leopard	(10)	(10) (7.5)	(0) (-2.5)
tiger	(10)	(10) (7.5)	(0) (-2.5)
cat	(10)	(10) (7.5)	(0) (-2.5)
wolf	(8)	(8) (7.5)	(0) (-0.5)
fox	(2)	(2) (7.5)	(0) (5)
rabbit	(5)	(5) (7.5)	(0) (2.5)
fly	(5)	(5)	(0)

Veras & Collins 2016



EduClust

A Visual Education Platform for Teaching Clustering Algorithms

Johannes Fuchs, Christian Rohrdantz, Andreas Stoffel, Matthias Miller, Daniel Keim

Parameters

k: 3

Distance Measure: Euclidian

$$d_2(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

Navigation

Iteration # 8

Step # 15

Current Animation Speed 2

Current Step 15

Visualization

Average Distance: 81.719

Algorithm: k-means

Complexity Range: O(k×n×t)

Input: *k* clusters

Output: *k* clusters

Pseudocode:

```
1. Choose k objects as initial cluster centers.
2. Assign each data point to the cluster which has the closest mean point (centroid) under chosen distance metric.
3. When all data points have been assigned, recalculate the positions of k centroids (mean points).
4. Repeat steps 2 and 3 until the centroids do not change any more. All data points remain in their most recently assigned cluster.
```

http://educlust.dbvis.de/

K-MEANS

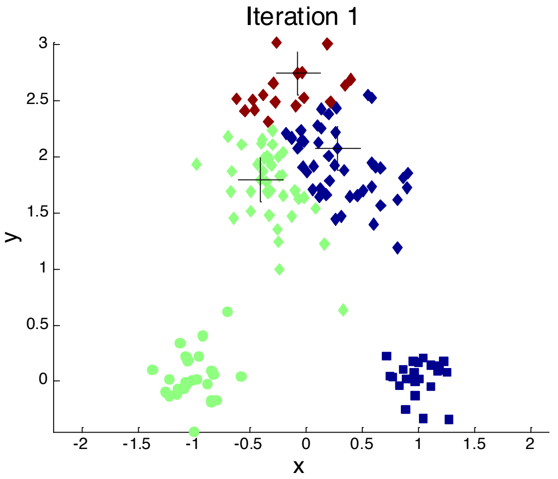
K-MEANS: THE OTHER MASSIVELY POPULAR CLUSTERING METHOD

Partitional clustering approach

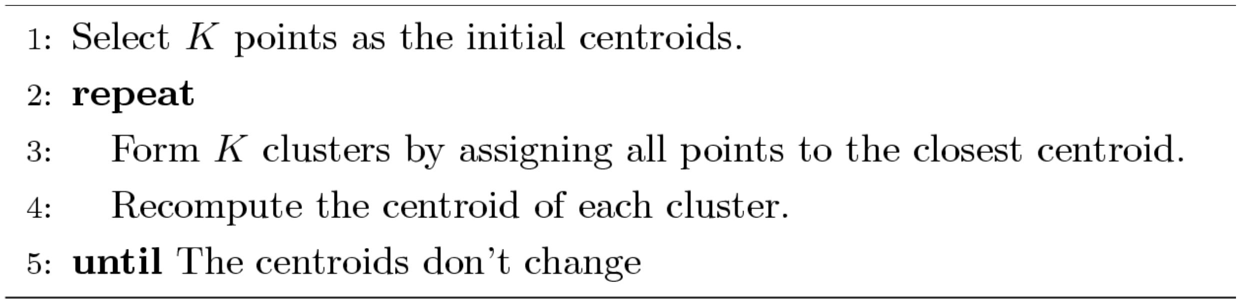
Each cluster associated with a **centroid** (center point)

Each point is assigned to the cluster with the closest centroid

Number of clusters, **K**, must be specified in advance

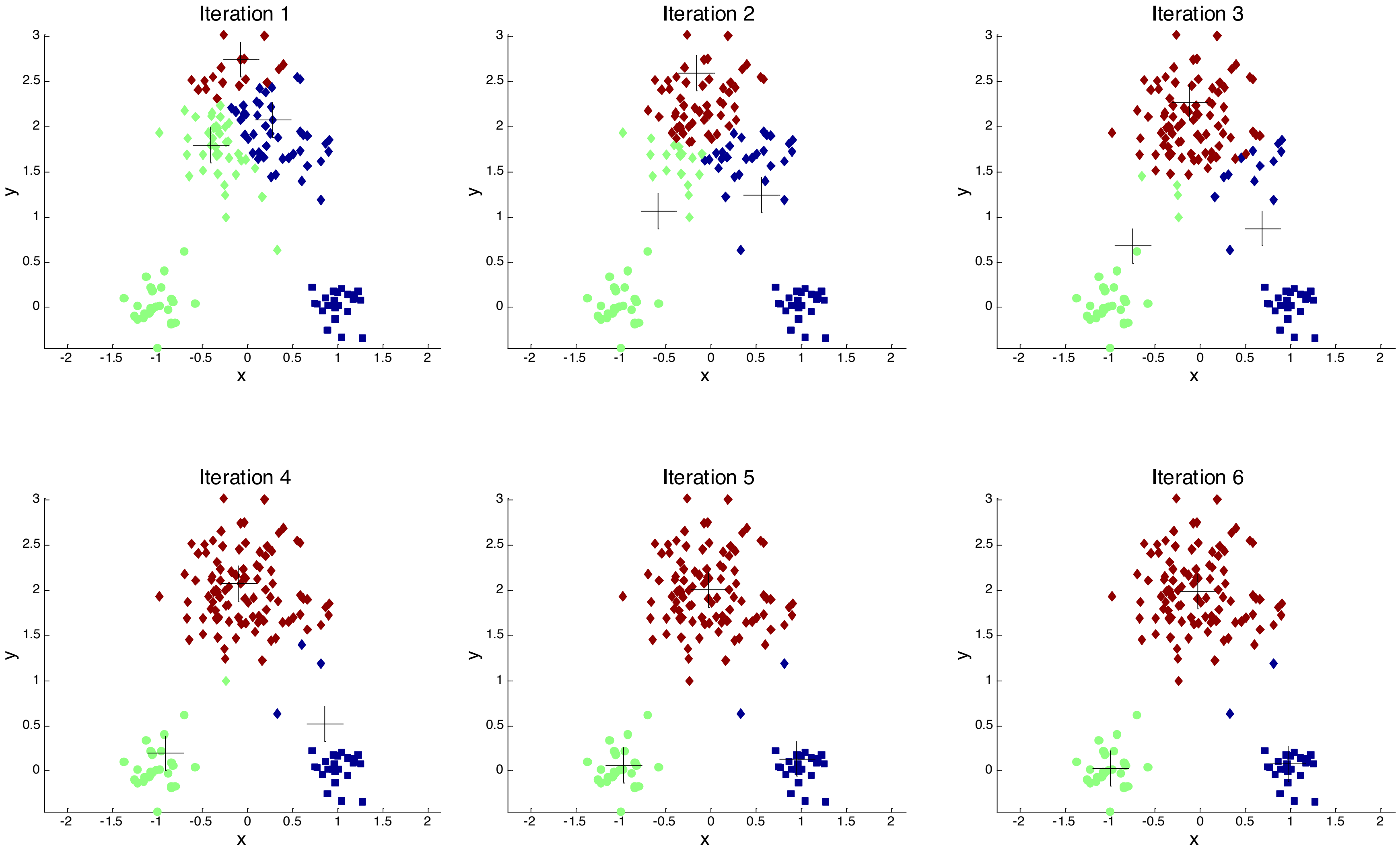


The basic algorithm is very simple



```
Select $K$ points as the initial centroids.
**repeat**
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
**until** The centroids don't change
```

THE K-MEANS ALGORITHM (K = 3)



Iteration 1

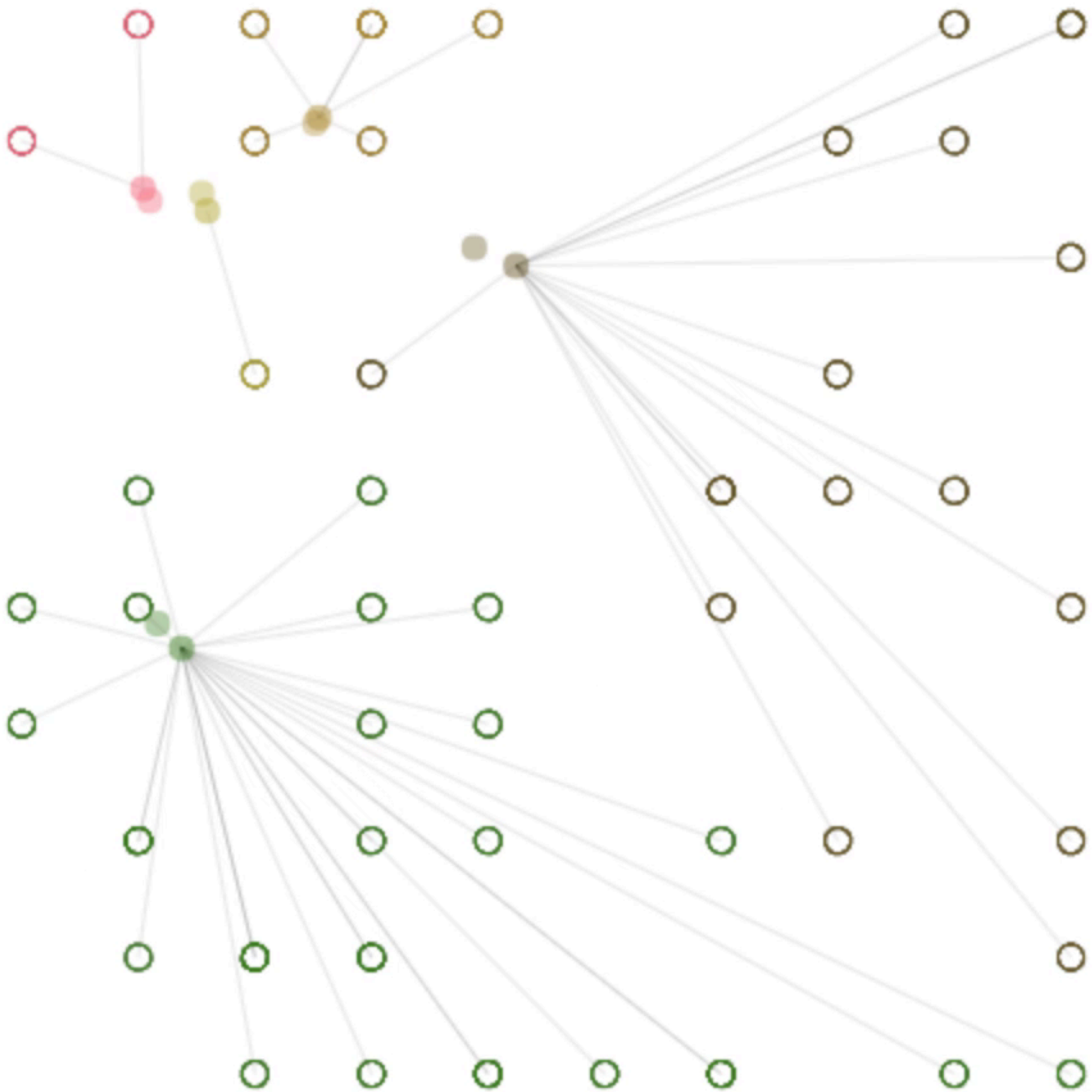
Iteration 2

Iteration 3

Iteration 4

Iteration 5

Iteration 6



Miguel Mota

https://goo.gl/LqhNUz

K-MEANS CLUSTERING - DETAILS

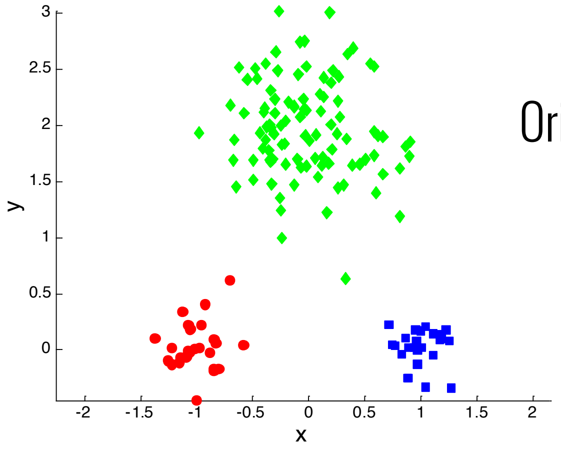
Different initializations can result in different solutions

- Initial centroids are often chosen randomly.
- Clusters produced vary from one run to another.
- So multiple runs are sometimes done

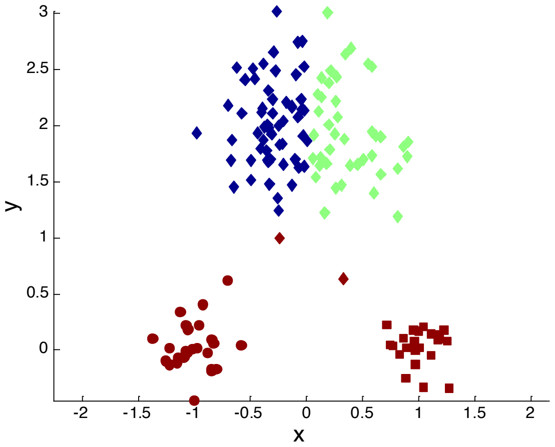
Centroid is typically the mean of the points in the cluster.

- “K-medioid” - center must be an actual datapoint.
- Useful when mean of a feature is not defined or available.

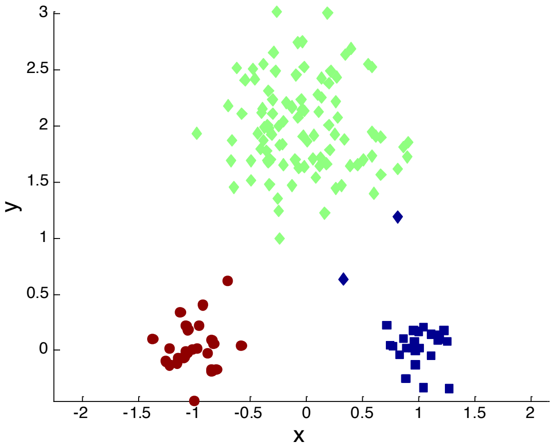
TWO DIFFERENT K-MEANS CLUSTERINGS



Original Points



Optimal Clustering? No.



Optimal Clustering

APPROACHES FOR IMPROVING K-MEANS

Idea 1: Be careful about where you start

Place first center on randomly chosen datapoint
Place second centroid on datapoint **as far as possible** from the first
Place *n*-th center on datapoint as far as possible from centers 1 thru $n - 1$

Idea 2: Do many runs of k-means

Each from a different random start configuration.
Use a heuristics to pick the best one.

LIMITATIONS OF K-MEANS

K-means has problems when clusters are of differing

- Sizes
- Densities

K-means has problems when the data contains **outliers**.

You **have** to pick the number of clusters (k) in advance.

WHEN TO USE K-MEANS VS HIERARCHICAL?

Do you need to easily **interpret** the clusters?

Do you **know the right** K ?

Does the data have a **natural "tree" structure** (living things, etc.)

How **computationally expensive** will each approach be for your data?

WHEN TO USE K-MEANS VS HIERARCHICAL?

k-means prefers solutions with similar sized clusters

- very different cluster sizes, shapes, densities can confuse it
- complex cluster geometry, or outliers
- need to specify and test for good *k* choice

Can combine the two approaches – for example:

1. Try several hierarchical methods and see which gives the most interpretable clusters.
2. Use k-means (with the hierarchical cluster centroids as starting points) to clean up the hierarchical cluster.

SUMMARY: CONDUCTING CLUSTER ANALYSIS

Formulate the Problem

Select a Distance/Similarity Measure

Select a Clustering Procedure

Decide on the Number of Clusters

Interpret and Profile Clusters

Assess the Validity of Clustering

HOW MANY CLUSTERS?

Theoretical, **conceptual** or **practical** issues may suggest a number.

Hierarchical clustering:

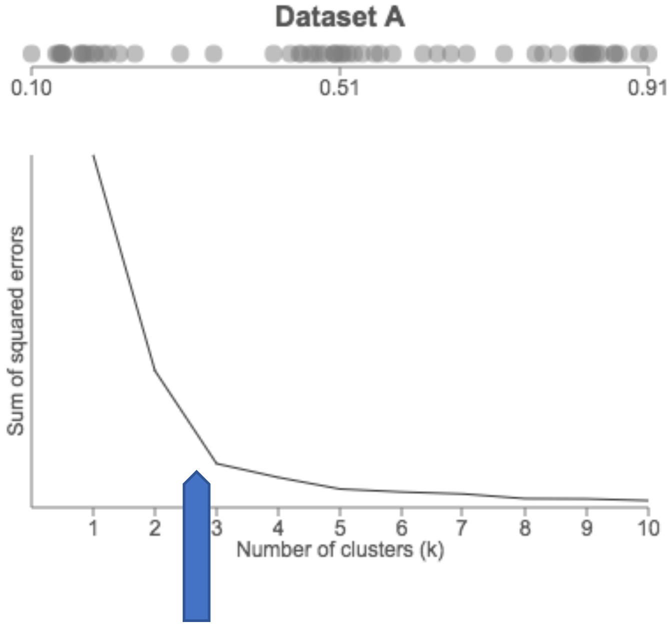
Distance threshold at which clusters are combined

K-means (and other non-hierarchical)

Compare ratio **within-groups/between-group variance** against the **number of clusters**

THE "ELBOW" METHOD

Compare **within-groups sum of squares** vs **# of clusters**



Example from Robert Gove

The "elbow" shows the point at which adding more clusters helps reduce distortion measure less and less.

SET RULES FOR SPLITTING / MERGING

ISODATA Algorithm

A **K-means** variant that can adds or remove centroids at each step based on user-defined thresholds like:

- Cluster size
- Standard deviation within cluster
- Distance between clusters
- Etc.

SUMMARY: CONDUCTING CLUSTER ANALYSIS

Formulate the Problem

Select a Distance/Similarity Measure

Select a Clustering Procedure

Decide on the Number of Clusters

Interpret and Profile Clusters

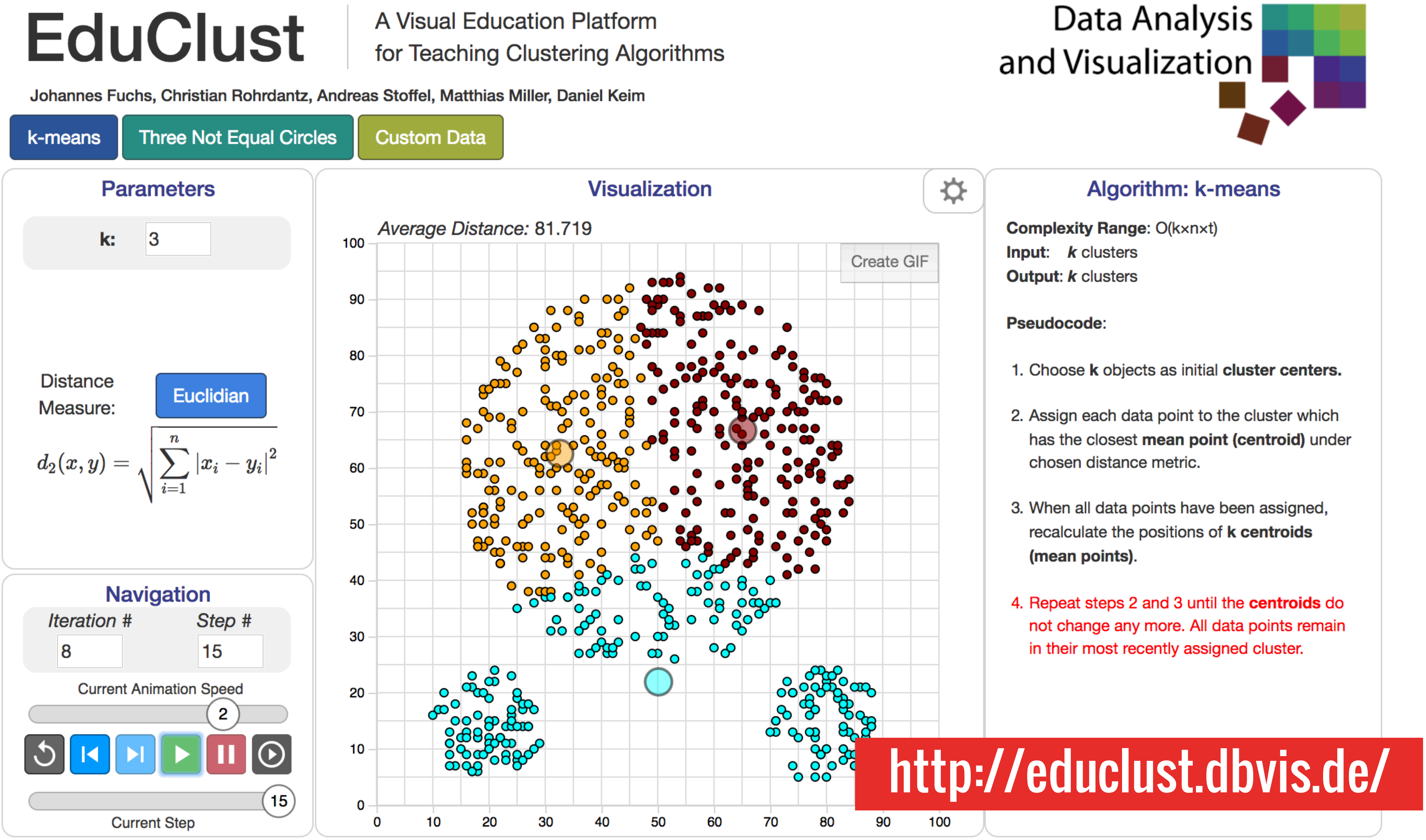
Assess the Validity of Clustering

HOW TO TELL IF YOU'VE FOUND GOOD QUALITY CLUSTERS?

Compare cluster stability across:

- Different **distance measures**
- Different **clustering methods**
- Different 50/50 **random data splits** (a bit like cross-validation)
- Different **variable/features deletions**
- Different **data orderings** (non-hierarchical)

"Good" clusterings (if they exist) are generally **stable** and **robust** to perturbations in methods or data.



EduClust

A Visual Education Platform for Teaching Clustering Algorithms

Johannes Fuchs, Christian Rohrdantz, Andreas Stoffel, Matthias Miller, Daniel Keim

Data Analysis and Visualization

Parameters

k: 3

Distance Measure: Euclidian

$$d_2(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

Navigation

Iteration # 8

Step # 15

Current Animation Speed 2

Current Step 15

Visualization

Average Distance: 81.719

Algorithm: k-means

Complexity Range: O(k×n×t)
Input: *k* clusters
Output: *k* clusters

Pseudocode:

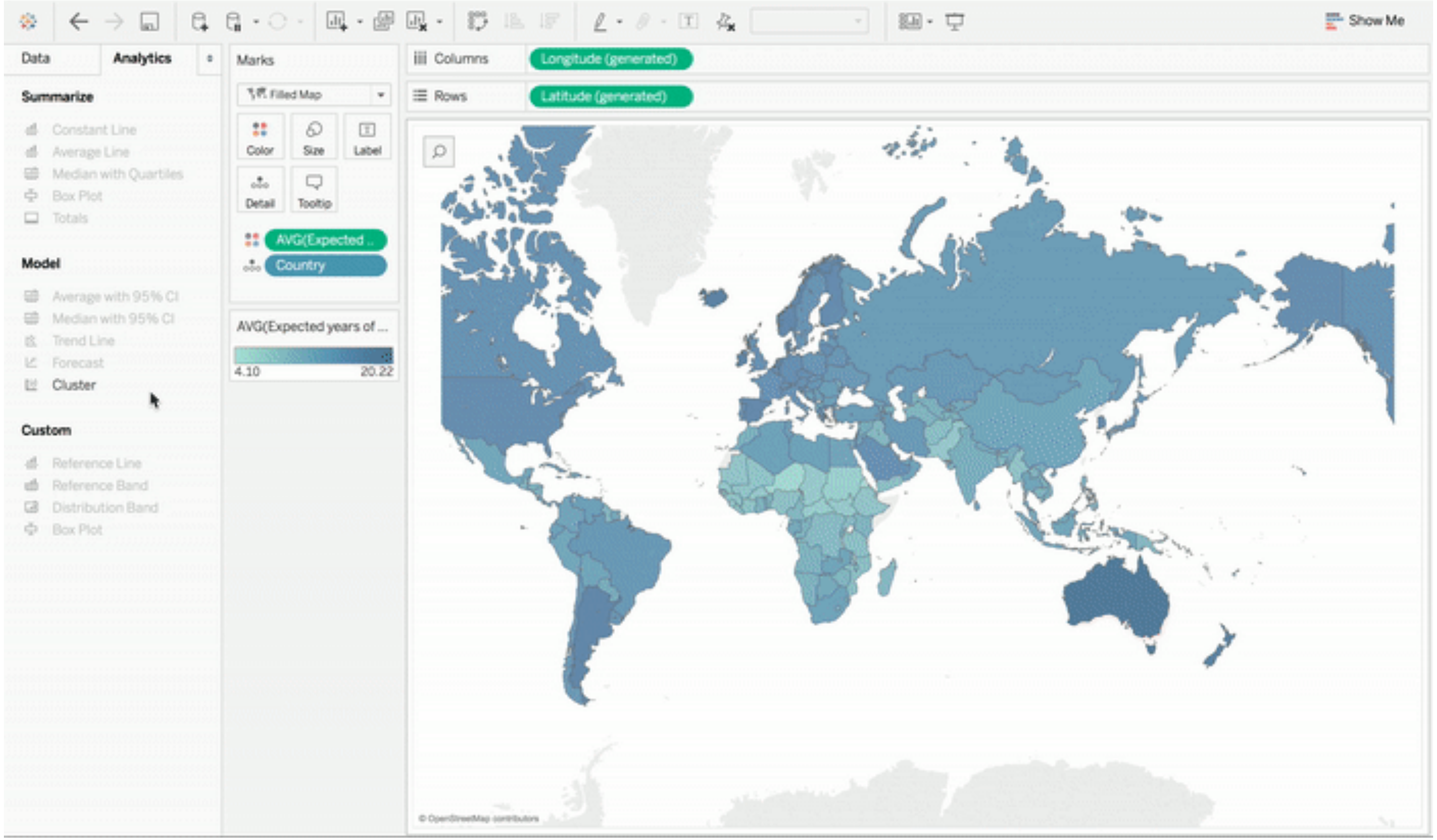
1. Choose **k** objects as initial **cluster centers.**
2. Assign each data point to the cluster which has the closest **mean point (centroid)** under chosen distance metric.
3. When all data points have been assigned, recalculate the positions of **k centroids (mean points)**.
4. Repeat steps 2 and 3 until the **centroids** do not change any more. All data points remain in their most recently assigned cluster.

http://educlust.dbvis.de/

CLUSTERING IN PRACTICE

Lots of good **Python and R libraries.**
Demos coming up!

K-MEANS IN TABLEAU



Columns Longitude (generated)

Rows Latitude (generated)

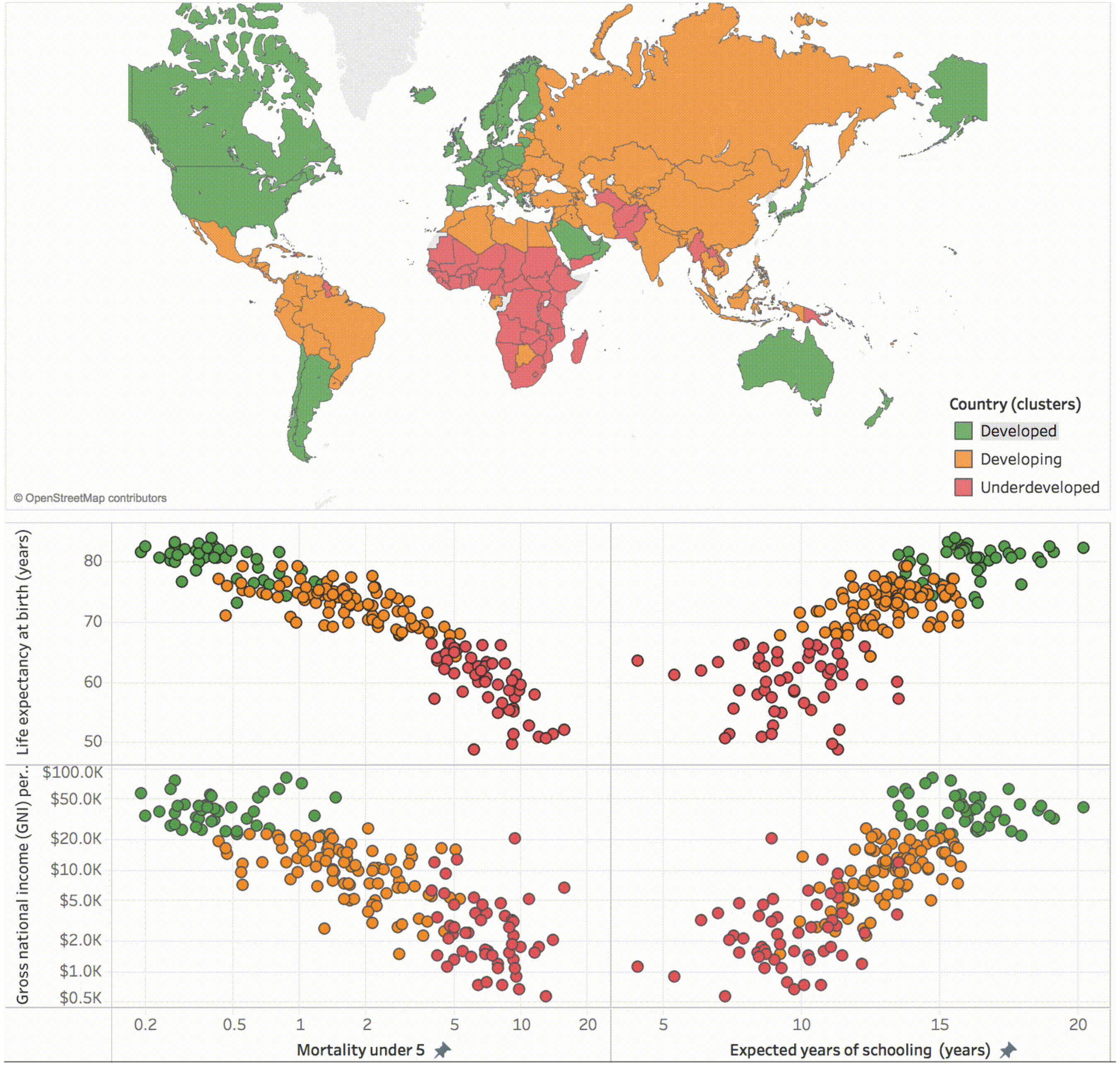
Marks Filled Map

AVG(Expected ..

Country

AVG(Expected years of ...

4.10 20.22



Country (clusters)
🟩 Developed
🟧 Developing
🟥 Underdeveloped
© OpenStreetMap contributors
Life expectancy at birth (years)
80
70
60
50
Gross national income (GNI) per..
$100.0K
$50.0K
$20.0K
$10.0K
$5.0K
$2.0K
$1.0K
$0.5K
0.2
0.5
1
2
5
10
20
Mortality under 5
5
10
15
20
Expected years of schooling (years)

SUMMARY

Clustering is a powerful and broad technique

Many, **many options** for each step

- Similarity metrics
- Type of clustering
  - Hierarchical, k-means, etc.
  - Decisions within each type
- Number of clusters

Many approaches **specifically for text (coming next week)**