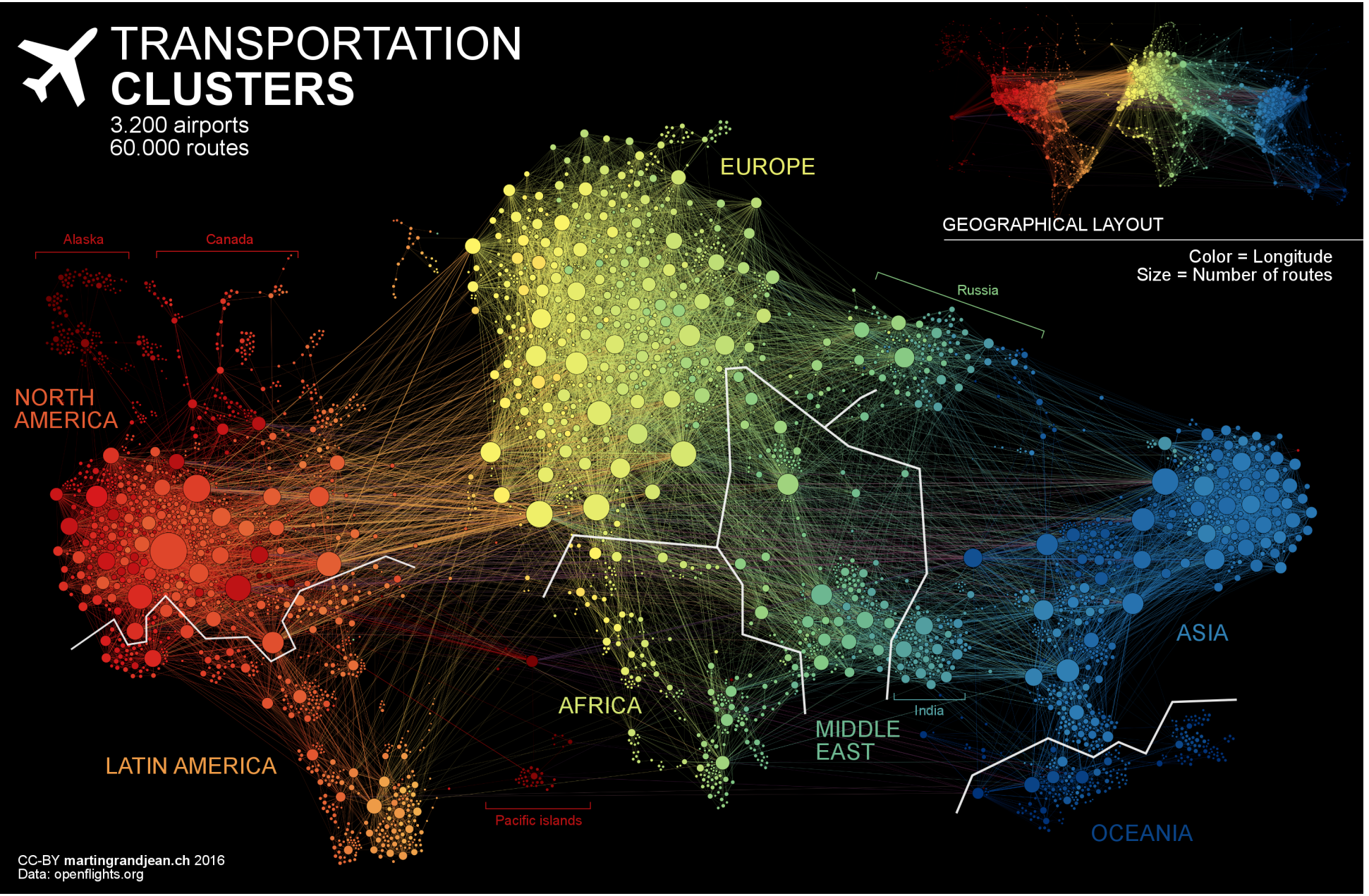
Hierarchical clustering from scratch

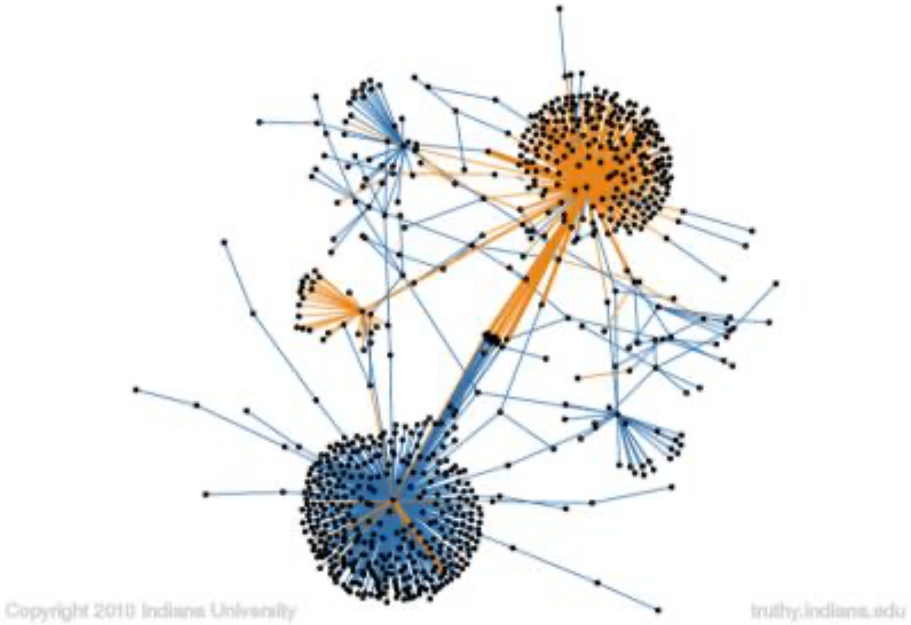
What is the problem?

We have a bunch of data points, and a way to measure the distance between them.

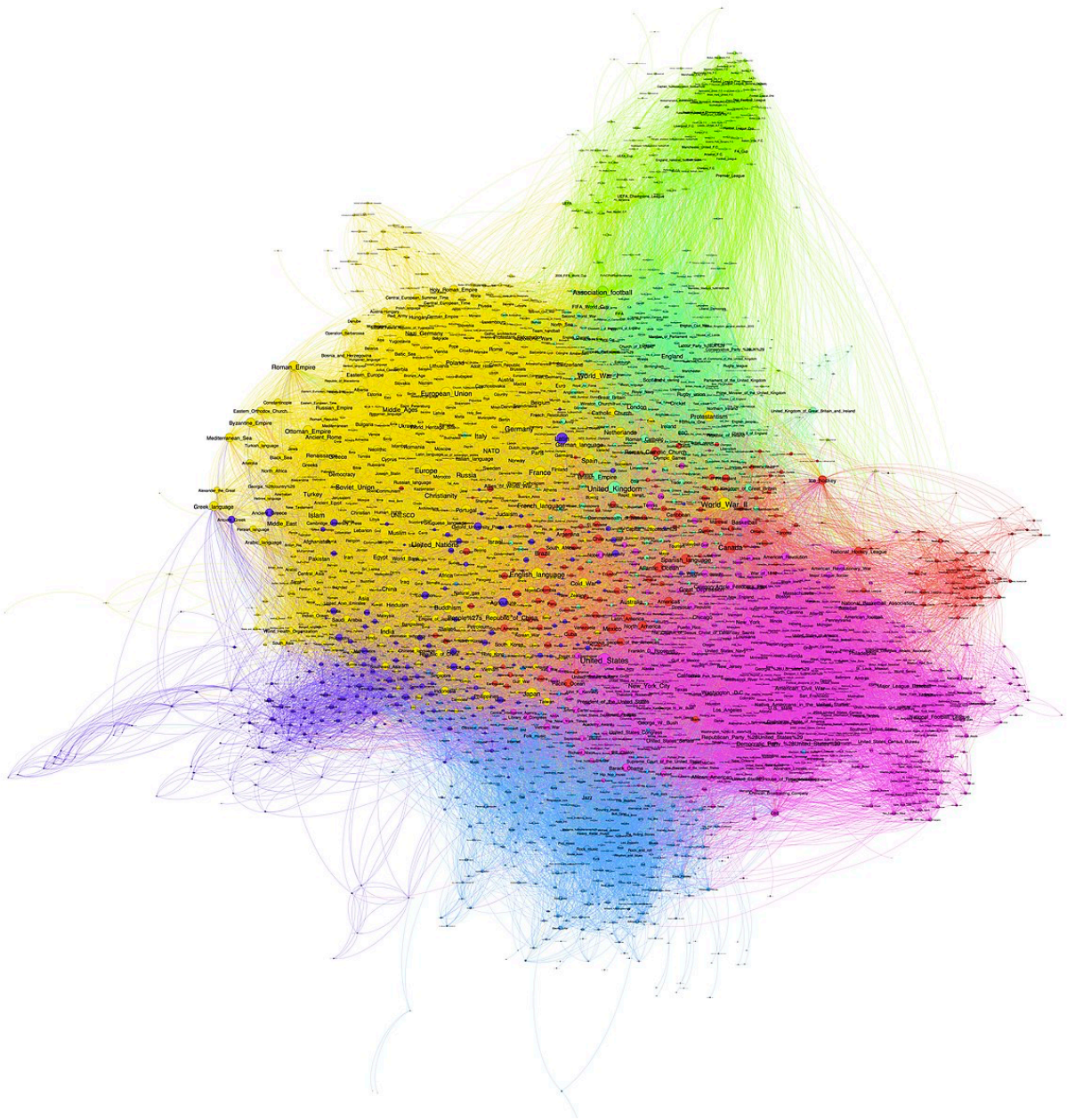
We want to know if our data points fall into interpretable groups—clusters.

Do people care about this sort of thing? Let's see some examples

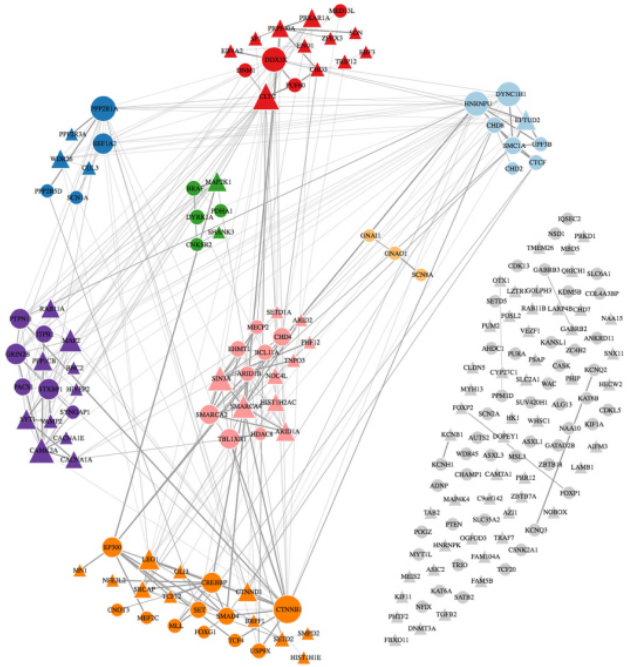

TRANSPORTATION
CLUSTERS
3.200 airports
60.000 routes
EUROPE
GEOGRAPHICAL LAYOUT
Color = Longitude
Size = Number of routes
Alaska
Canada
Russia
NORTH
AMERICA
ASIA
AFRICA
India
MIDDLE
EAST
LATIN AMERICA
Pacific islands
OCEANIA
CC-BY martingrandjean.ch 2016
Data: openflights.org



Pictured here is a diffusion network created by Truthy.indiana.edu for the Twitter burst generated by Lady Gaga supporters toward John McCain following Gaga's comments about McCain's opposition to repealing Don't Ask, Don't Tell. The meme was tweeted 1,276 times by 1,100 users, with 168 users retweeting 696 times and another 59 users mentioning the meme 325 times. Links in orange are mentions; blue links show retweets.



The top 2500 Wikipedia pages (by number of internal links) clustered.
A guess at some of the clusters:
Purple = USA
Yellow = Europe, other countries Light green = UK
Mid green = Football
Blue = music
Dark purple = science
Red = non-USA non-European countries



Autism risk genes

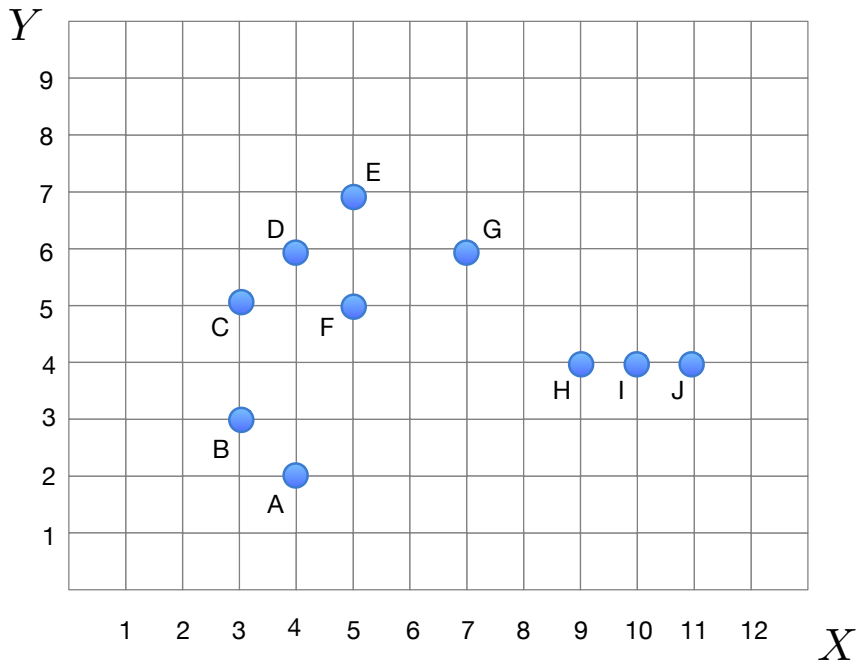
So it looks like cluster analysis is actually useful.

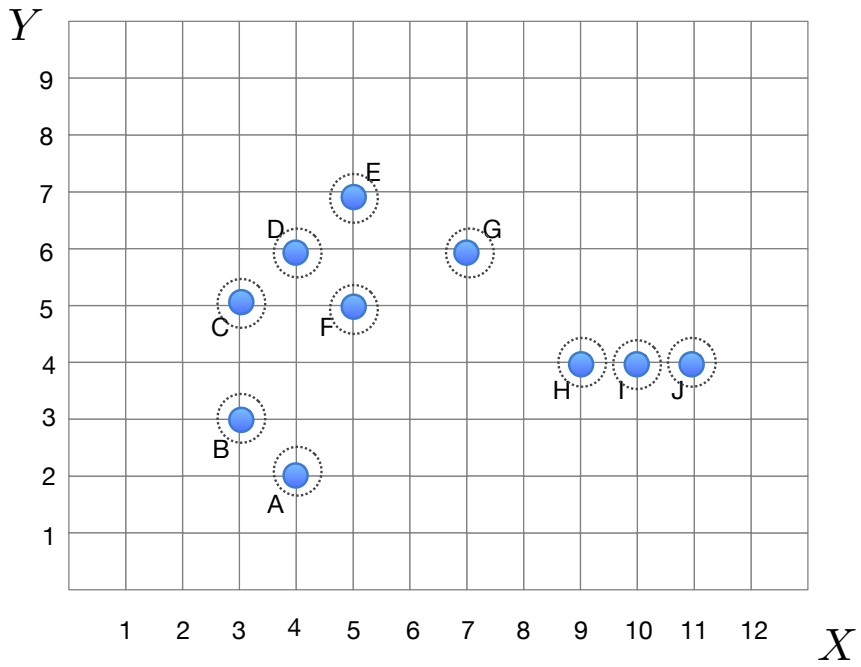
Hierarchical clustering from scratch

How does it connect with what we have studied so far?

1. **Linear regression**: we use independent variables to estimate the value of a dependent variable
2. **Logistic regression**: dependent variable is qualitative. We estimate a classification label
3. In both cases we are estimating some dependent variable
4. In **cluster analysis we are not looking at the relationshi**p between independent and dependent variables
5. We are seeing whether our data points are naturally grouped in ways we can interpret
6. Clustering falls into a class of unsupervised algorithms in which we mine patterns from data.

A simple example

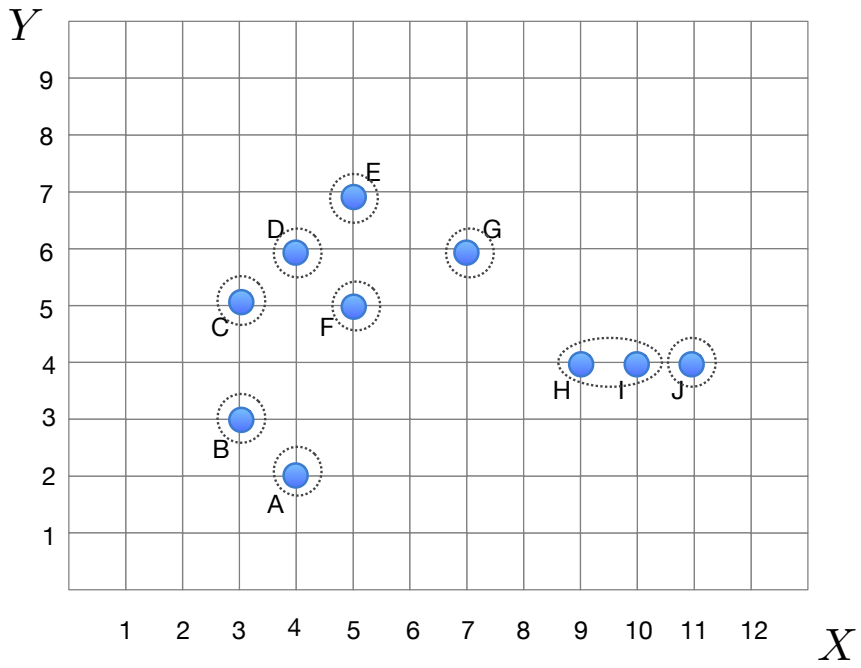




At the beginning, each data point is its own cluster?

What are the two clusters that are nearest?

Assume linkage = **single**.



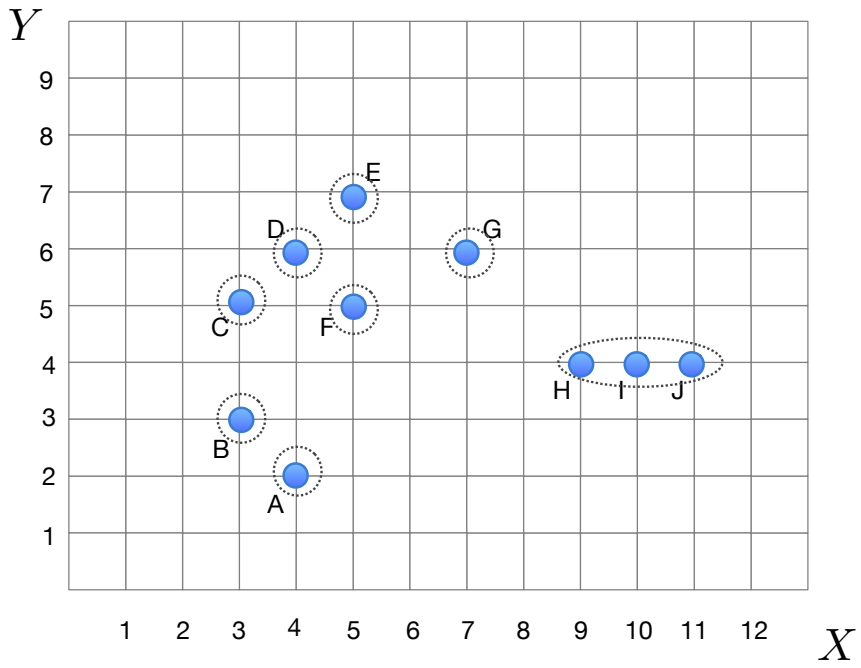
At the beginning, each data point is its own cluster?

What are the two clusters that are nearest? You have to compute the distance matrix.

$H - I$ and $I - J$

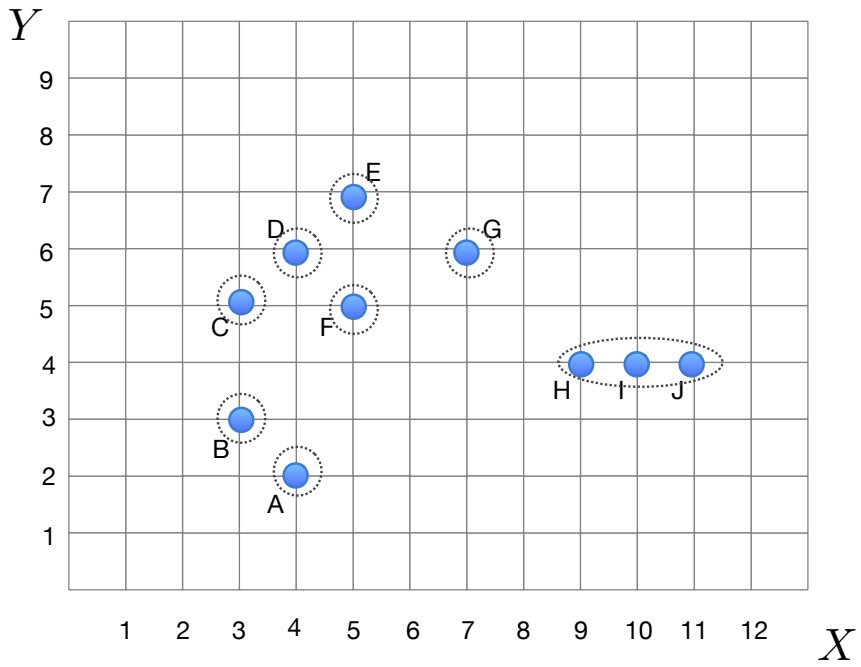
Pick any. Now we have nine clusters. Which two clusters are nearest to each other now?

How to we compute the distance since linkage is **single**?



HI and *J*

And what two clusters are closest to each other now?



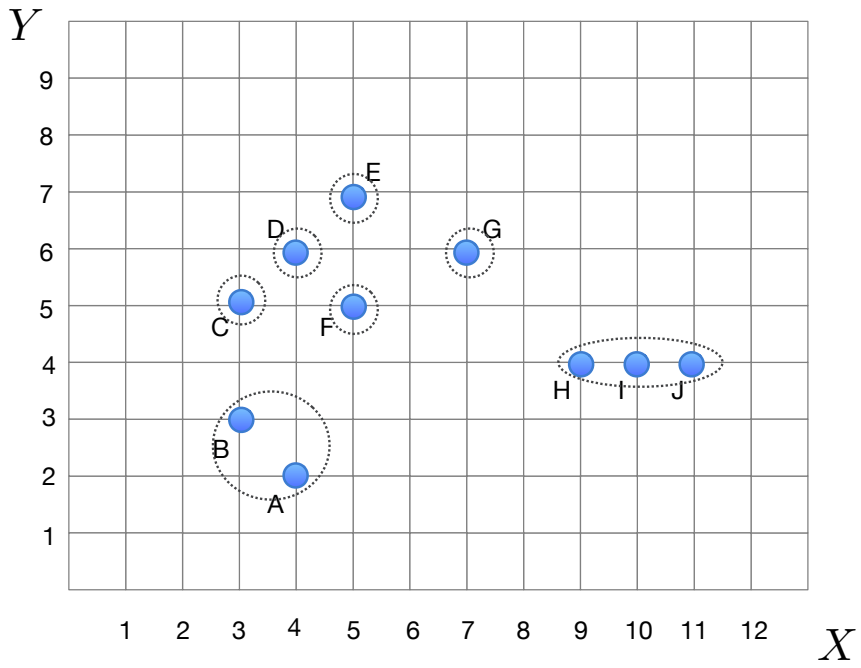
HI and J

And what two clusters are closest to each other now?

Quite a few.

$A - B, C - D, D - F, D - E$

What to pick?



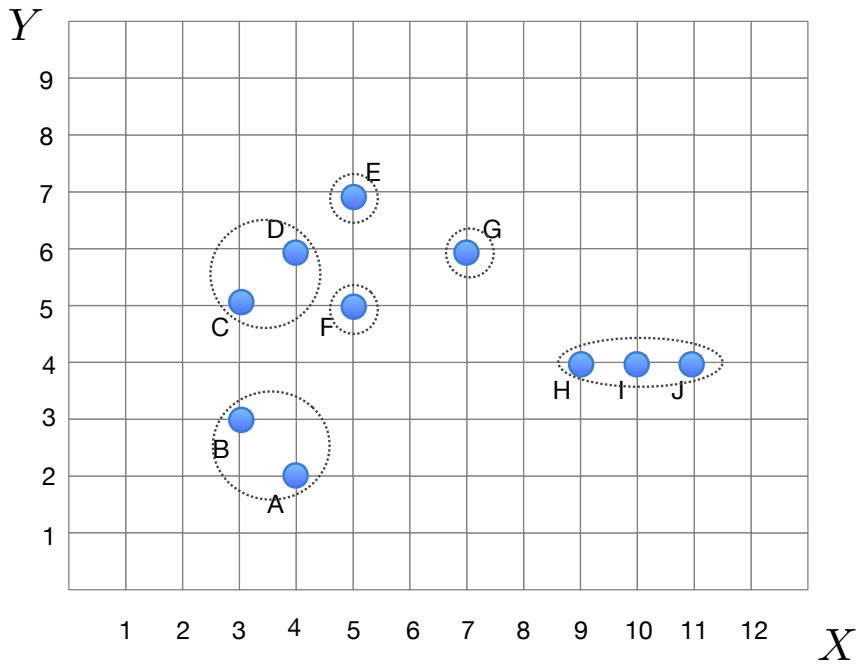
HI and J

And what two clusters are closest to each other now?

Quite a few.

$A - B, C - D, D - F, D - E$

What to pick?



HI and J

And what two clusters are closest to each other now?

Quite a few.

$A - B, C - D, D - F, D - E$

What to pick?

Create your own Python code from scratch to validate.

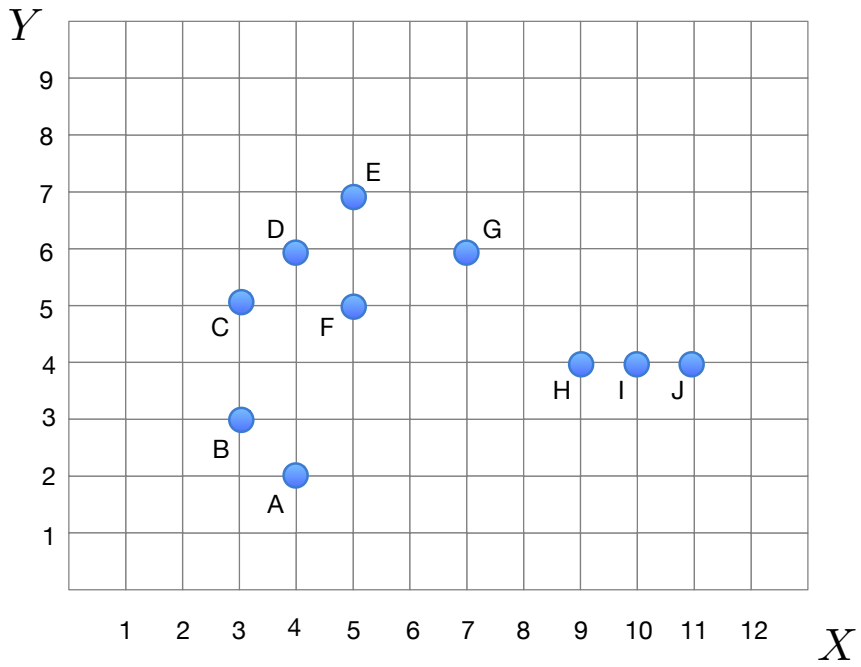
K-means clustering

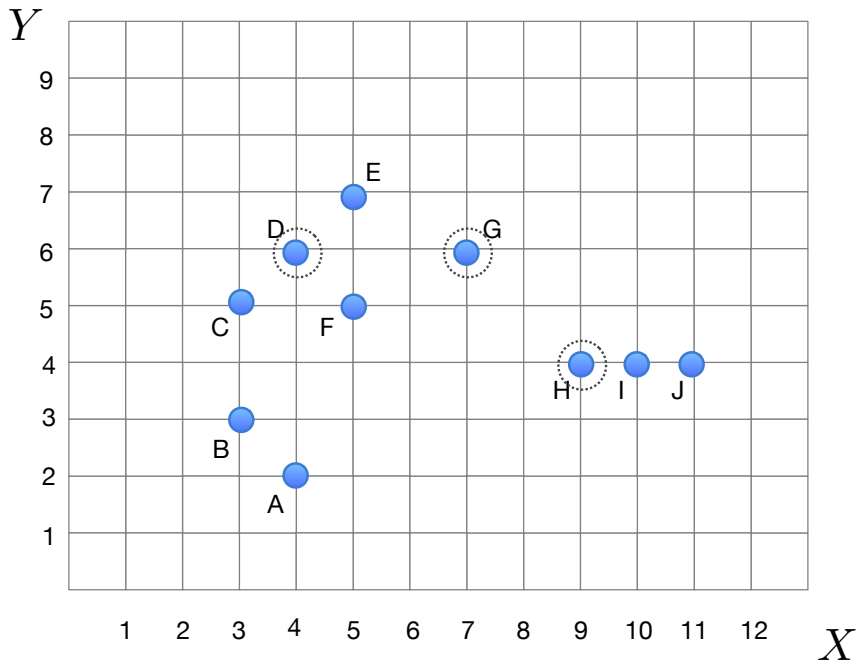
What is the problem?

We have a bunch of data points, and a way to measure the distance between them.

We want to partition the data points into k non overlapping clusters.

A simple example (again…)



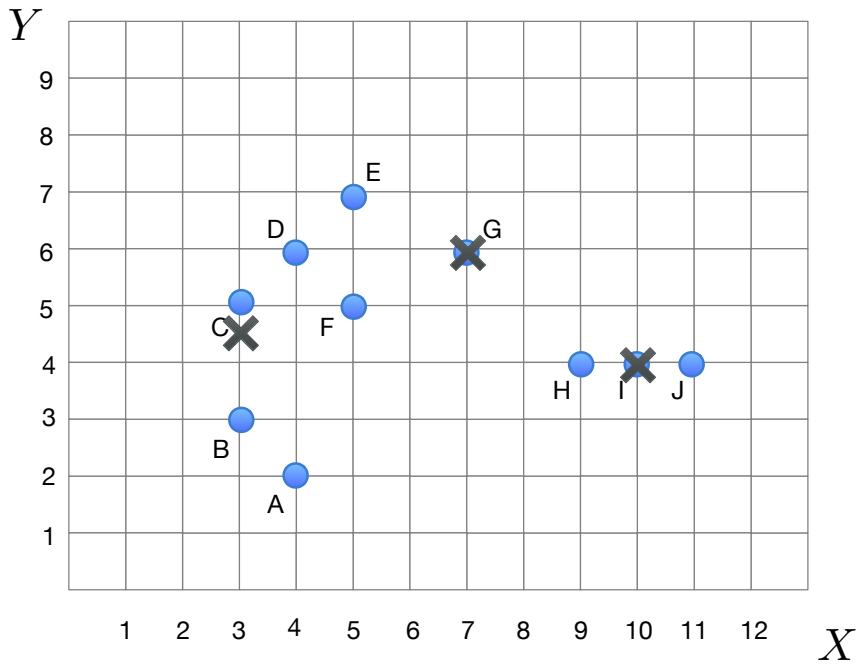


Select $k = 3$ random data points to be the initial cluster centroids.

Assign each remaining data point to the nearest centroid:

A -> D
B -> D
C -> D
E -> D
F -> D
I -> H
J -> H

Compute the new centroid for each cluster as the mean of all the points in the cluster



For D:

X coordinate
A: 4, B:3, C:3, D: 4, E:5, F:5
Mean = 4

Y coordinate
A: 2, B:3, C:5, D: 6, E:7, F:5
Mean = 4.66

For J:
(10,4)

For G:
No change

Now what?