Lecture 9: Unsupervised Learning

Intro to Data Science for Public Policy
Spring 2017

Jeff Chen + Dan Hammer

Roadmap

- Last thoughts on supervised learning
- Unsupervised learning
- K-Means
- <break>
- Hierarchical clustering

<short-story/>

Checklist for structuring your project

- [ ] 1) Do each Y=1 and Y=0 have at least n = 100?
- [ ] 2) Does either Y=1 or Y=0 fall below 10% of the sample?
- [ ] 3) Is your sample drawn from a population on which you can generalize and score?
- [ ] 4) Does your sample represent what you think it represents?

Checklist for structuring your project

- [ ] 1) Do each Y=1 and Y=0 have at least n = 100?

If not, get more data.

Checklist for structuring your project

☐ 2) Does either Y=1 or Y=0 fall below 10% of the sample?

If not, consider upsampling the small proportion by bootstrapping or duplicating the imbalanced class until the two classes have equal proportions. Consequence is that accuracies will be misleading.

For example: If Y = 1 (n = 300) but Y = 0 (n = 10000), then consider bootstrapping Y = 1 by 33x.

Checklist for structuring your project

☐ 2) Does either Y=1 or Y=0 fall below 10% of the sample?

Why? Because confusion matrices will likely be misleading due to the differences in proportions.

	Pred: T	Pred: F
Actual: T	0	300
Actual: F	0	9700

Accuracy = 97% with a TPR = 0% is not accurate.

Checklist for structuring your project

☐ 3) Is your sample drawn from a population on which you can generalize and score?

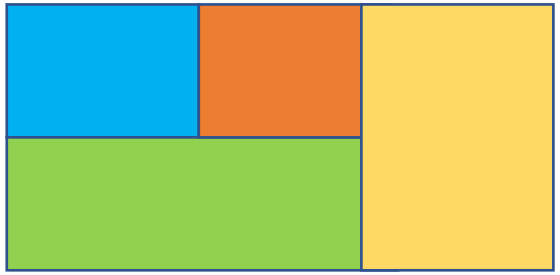
If your sample is a one time deal (e.g. a one time survey, a one time data release), make sure you're comfortable with your intended use.

For example: Fire prediction algorithms should have a constant stream of data.

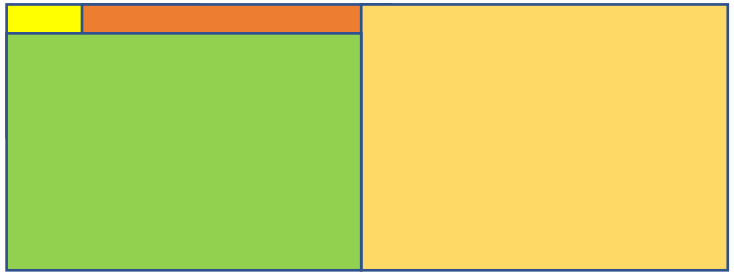
Checklist for structuring your project

☐ 4) Does your sample represent what you think it represents?

- Not all answers are in the data if the data does not have enough coverage.
- State your hypotheses. What constitutes the H0 and H1? Does your data have enough coverage to prove H1?



What you hope is in the data



Data doesn't cover all of the subpopulations you need.

Roadmap

- Last thoughts on classifiers
- Unsupervised learning
- K-Means
- <break>
- Hierarchical clustering

What if you didn’t have labels in a dataset?

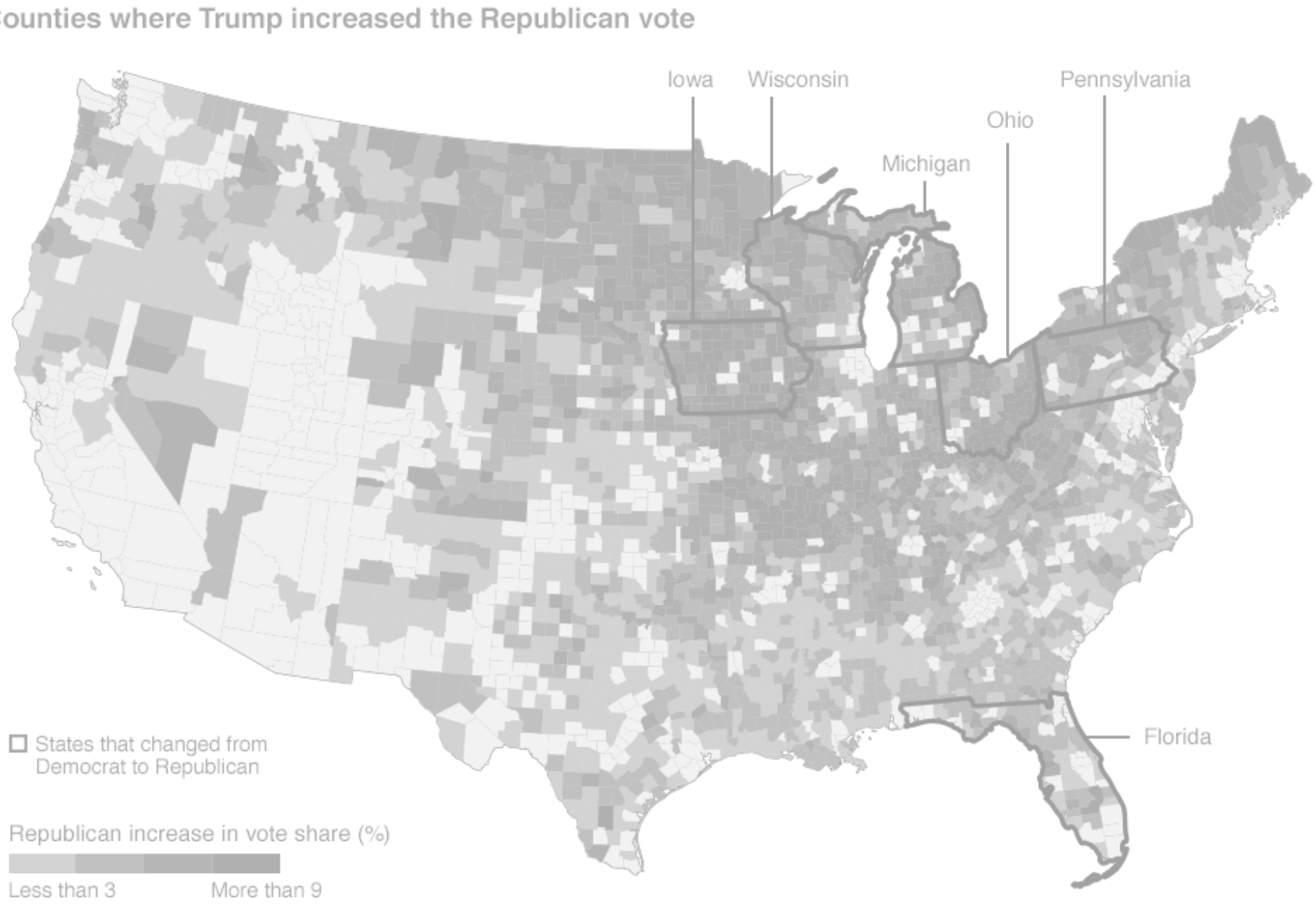
How much of the land is yielding growth?

Courtesy NASA

How can we extract the extent of the fertile areas of the Nile from a photograph?

Courtesy NASA

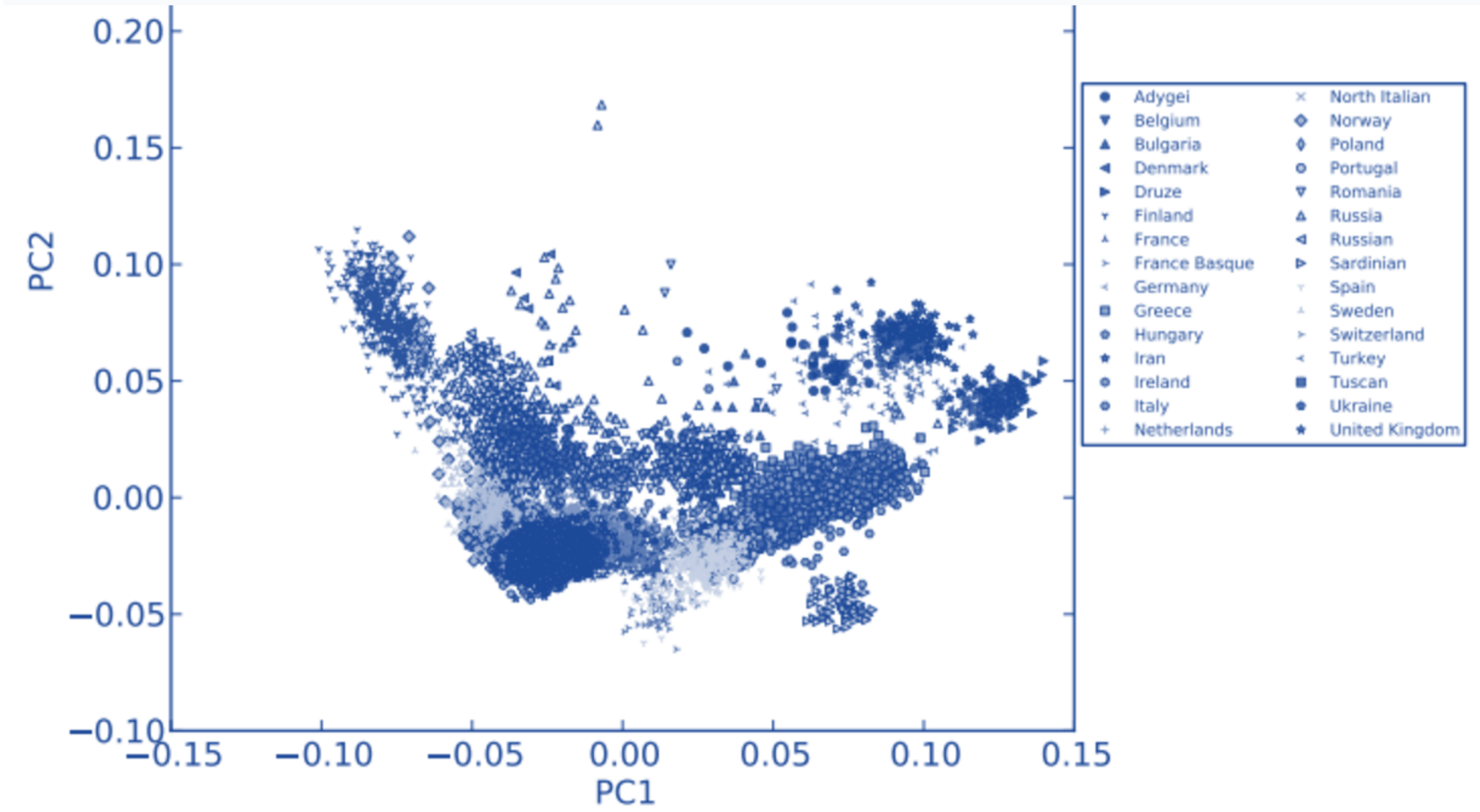
In the last election, which regions' support swung for one party more than before?



K-means

Credit: BBC

Which genes are correlated and can be expressed in lower dimensions?



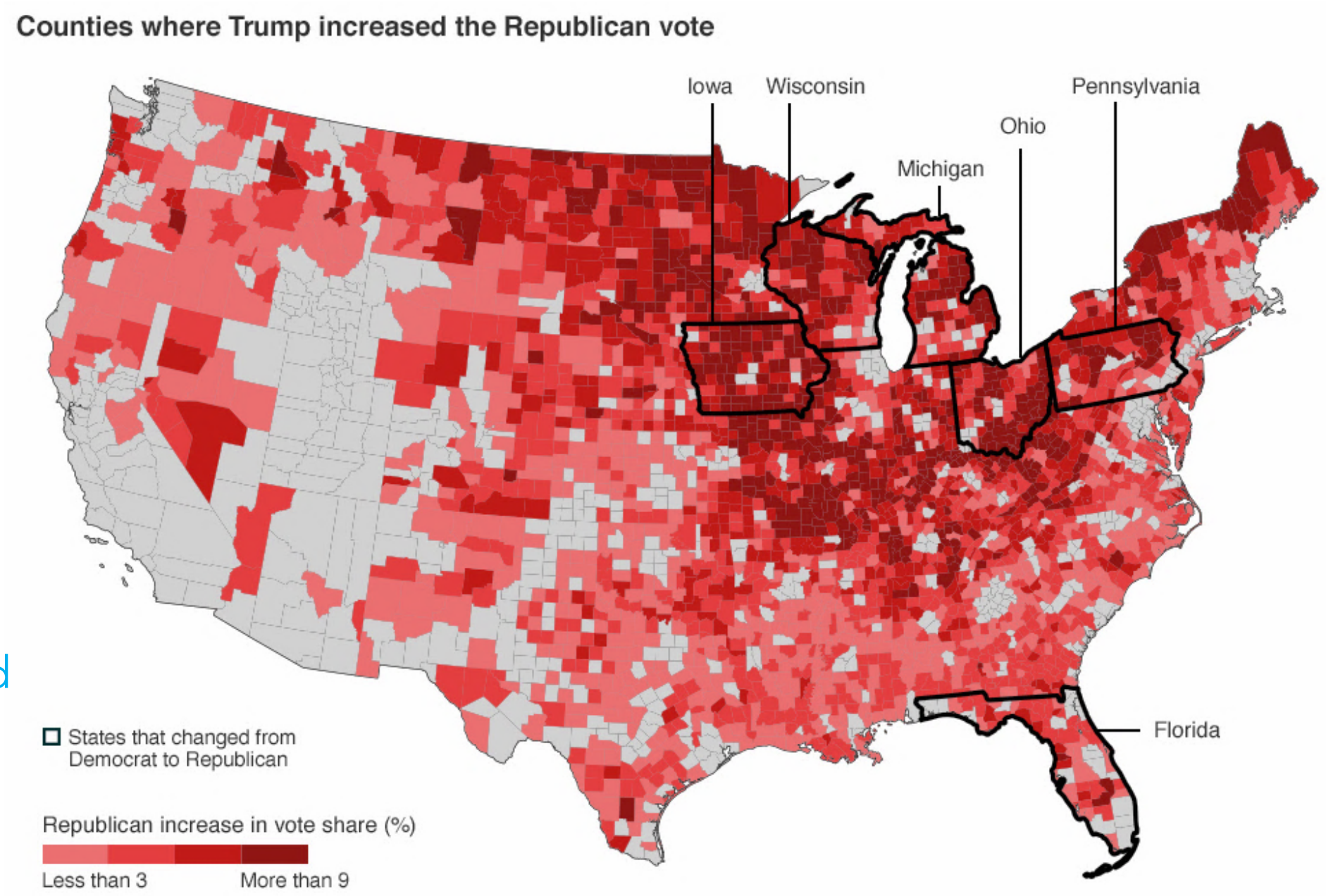
K-means Credit: 23andMe

Answers: A lot or 46.7% of the image.

woot! →

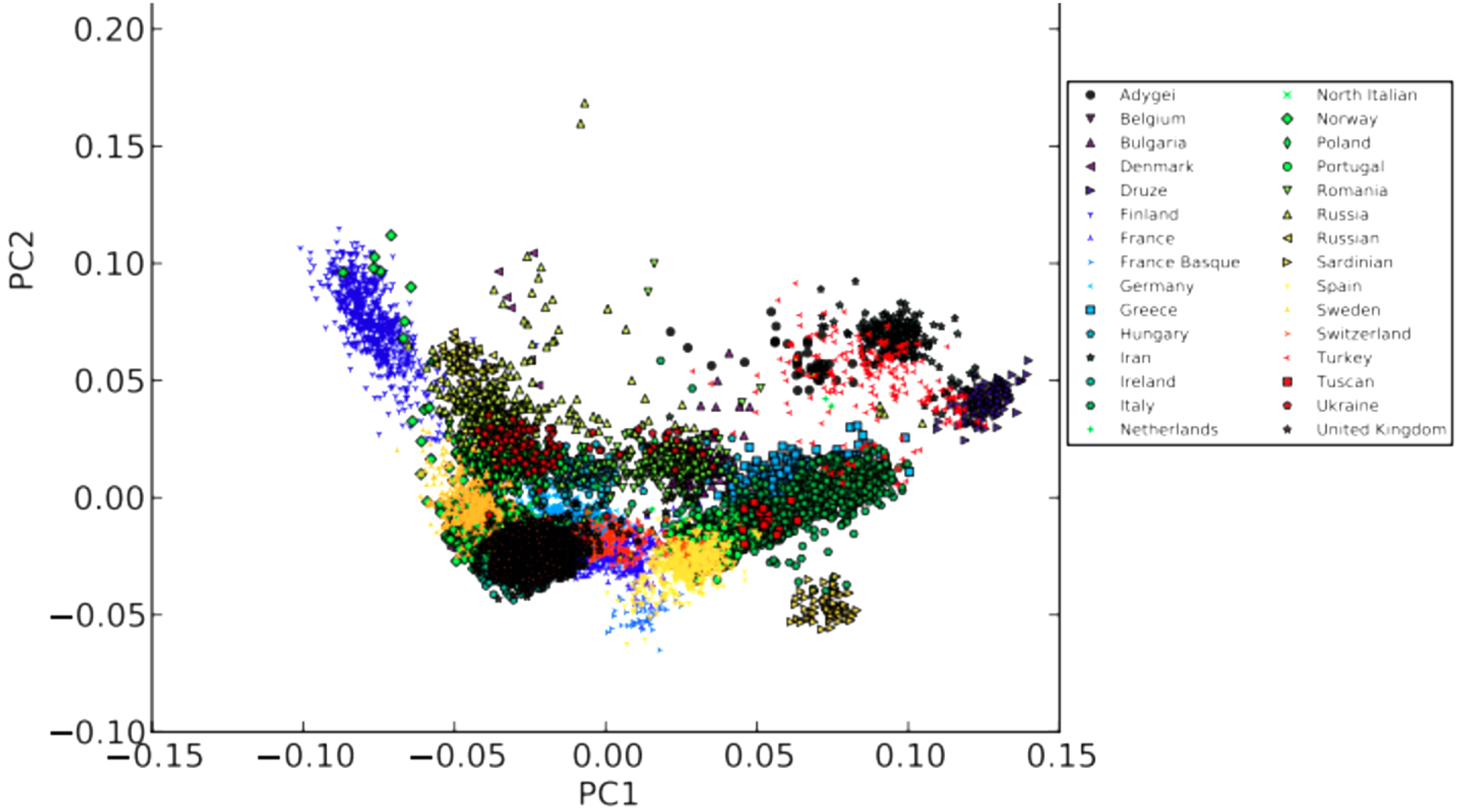
Which regions' support swung for one party more than before?

(Note that the pattern is not bound by states)



Credit: BBC

Depends. But for some ancestries, it can be as few as two variables



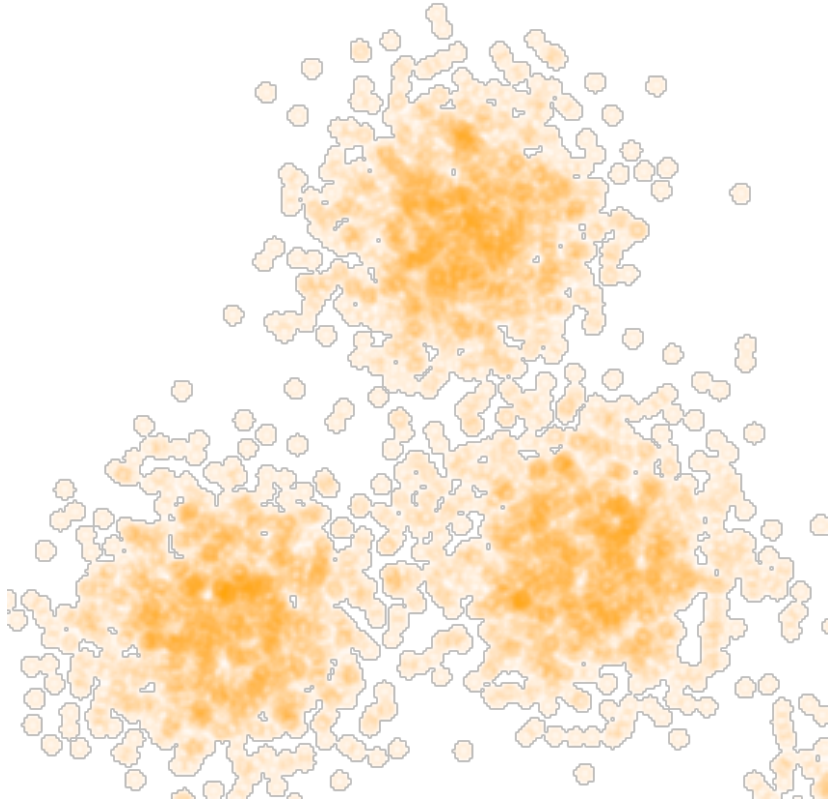
K-means Credit: 23andMe

Supervised Learning

- Goal: Learn and predict known phenomena
- Data: Targets (labels), Input features
- Eval: Common measures (e.g. ROC, F1, TPR)
- Complexity: Statistical techniques tend to be complex

$$y = f(x)$$

Unsupervised Learning



- Goal: Find new patterns via clustering and latent features
- Data: Only input features
- Eval: Many evaluation measures or techniques, but none that are standard
- Complexity: Statistical techniques tend to be simple and iterative

Two general tasks in unsupervised learning

Clustering can answer:

Which points can be grouped together based on their attributes?

Two common types:

- K-means
- Hierarchical Clustering (Agglomerative)

Dimensionality reduction

Which features have similar information and can be packaged into fewer principal variables

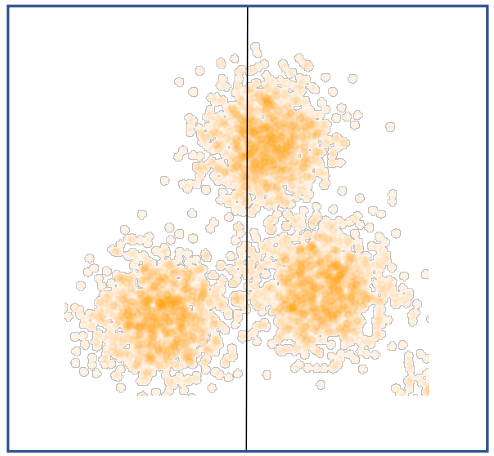
Two common types:

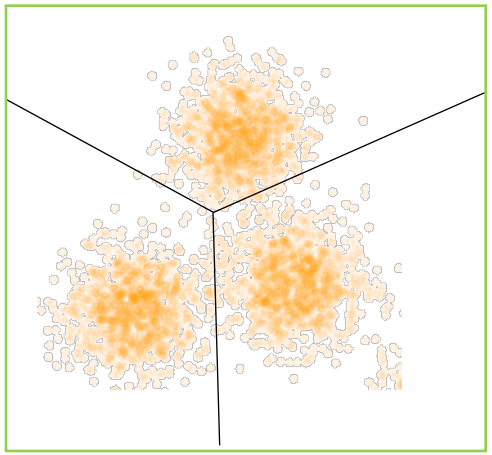
- Principal Component Analysis (PCA)
- Latent Factor Analysis

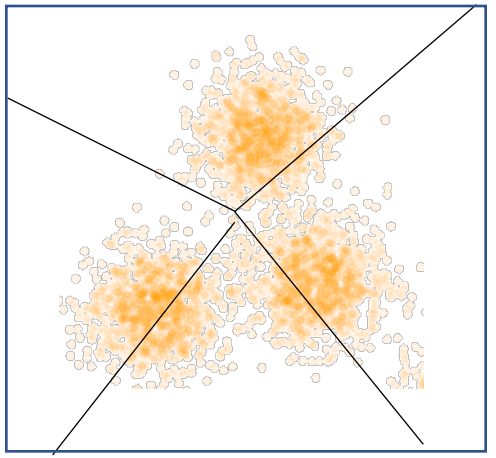
Formulation of Clustering Problems

Assumption:

There is some set of clusters k < n that represent naturally occurring groups



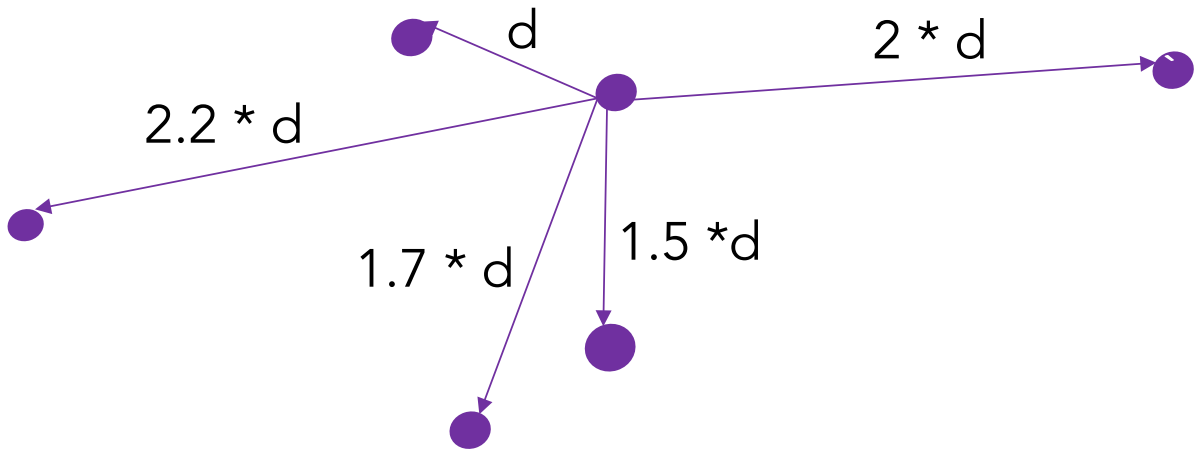




Formulation of Clustering Problems

Distance:

Some measure of distance based on input features can be used to determine similarity.



Formulation of Clustering Problems

Distance:

There are many types of distance. Selection depends on the type of problem and type of data.

Euclidean = $\sqrt{\sum(x_{ik} - x_{0k})^2}$

$Manhattan = |x_{ik} - x_{jk}|$

$$\cos(\theta) = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}}$$

Formulation of Clustering Problems

Stability:

Convergence may occur in a model run, but stability needs to be tested.

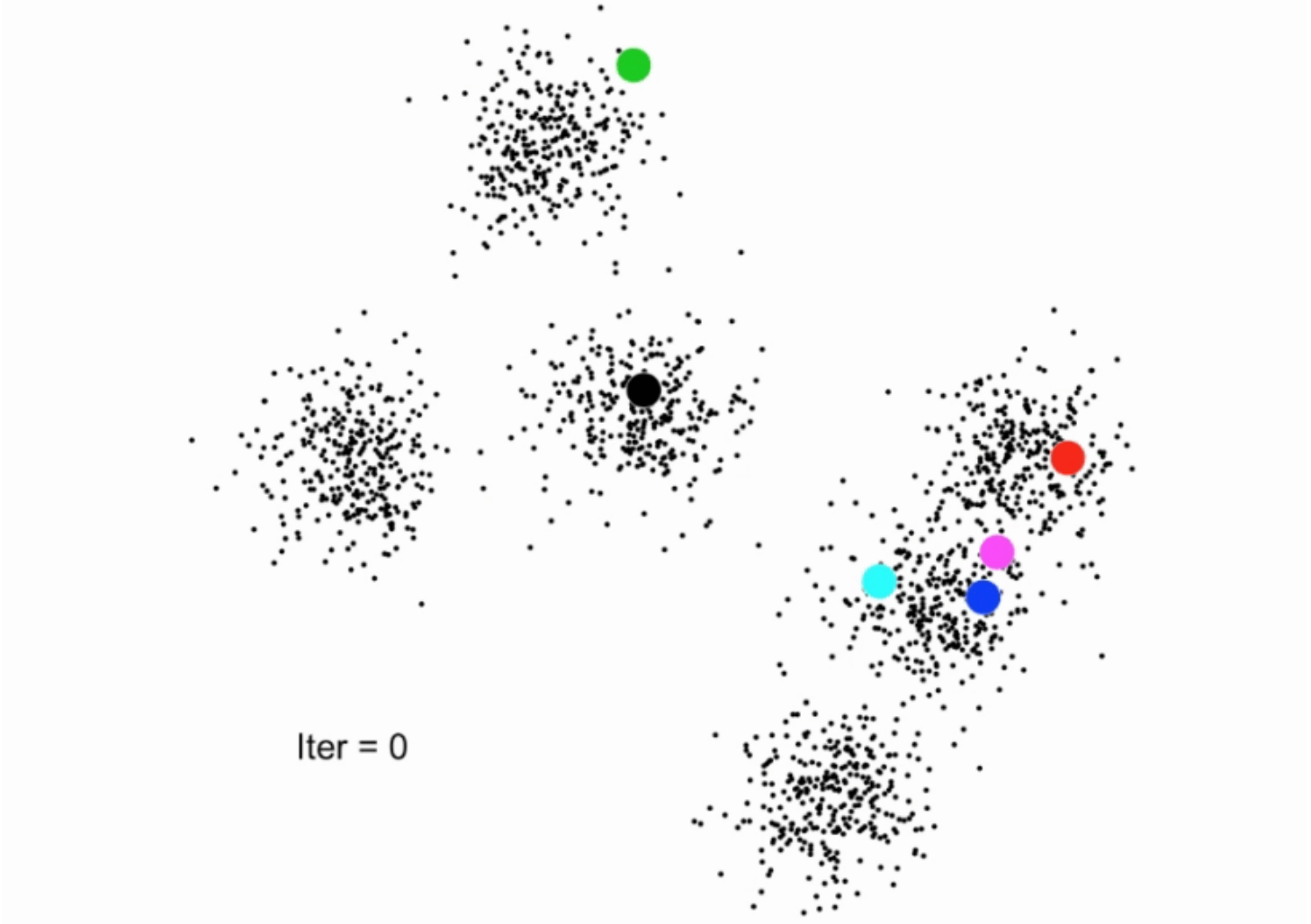
Roadmap

- Last thoughts on classifiers
- Unsupervised learning
- K-Means
- <break>
- Hierarchical clustering

Common uses of clustering

- Marketing/Sales
  - Customer/audience segmentation – which characteristics of people tend to buy what? How do demand segments differ?
- Genetics:
  - Clustering gene sequences and expressions
- Image Pattern Detection:
  - Extracting features from images
- Demography
  - Identifying demographic enclaves

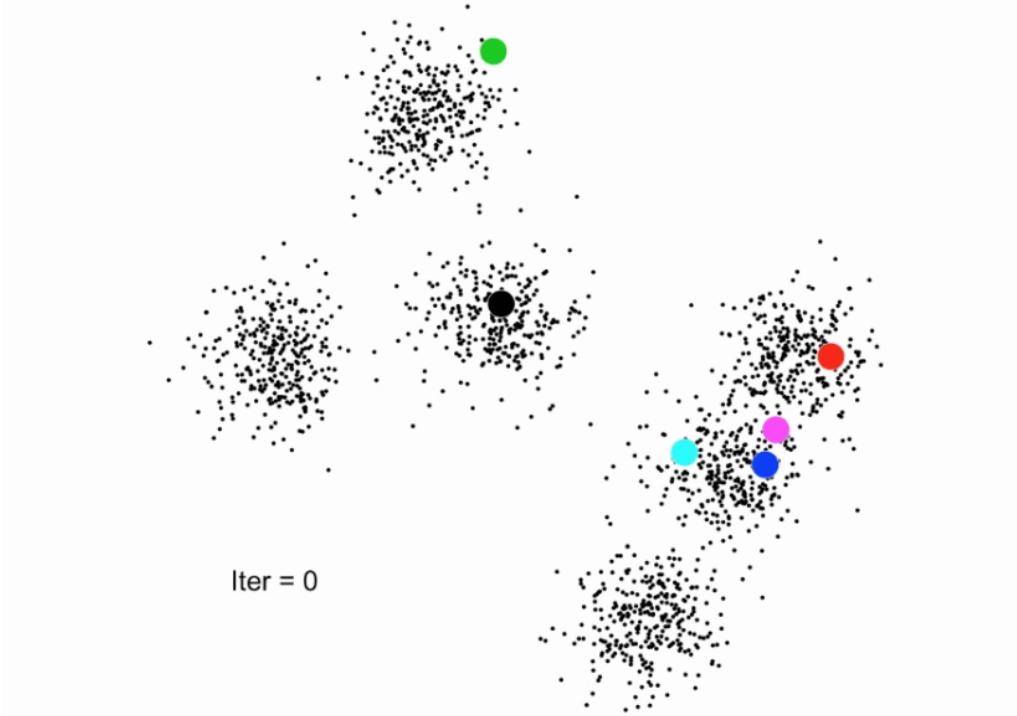
What is k-means in 10 seconds?



In short

Given k-number of user-specified clusters, K-Means finds the local optima of a dataset such that each point is assigned to a cluster

An algorithm that is as simple as it gets




```
Initialize k centroids
Repeat until convergence:
  Calculate distance between each record n and centroid k
  Assign points to nearest centroid
  Update centroid coordinates as average of each feature per cluster
```

How it works

$$argmin \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_{i,j} - \mu_j||^2$$

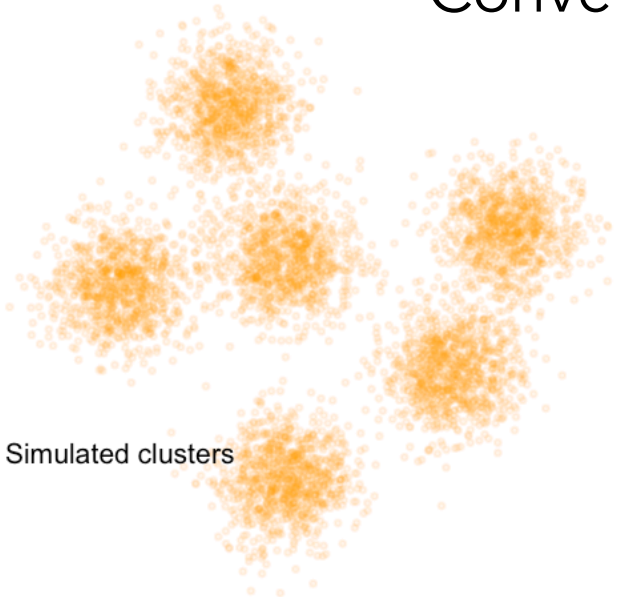
The goal is the minimize (arg min) the within-cluster variance (||x – u||^2), which is calculated for each point i and cluster j.

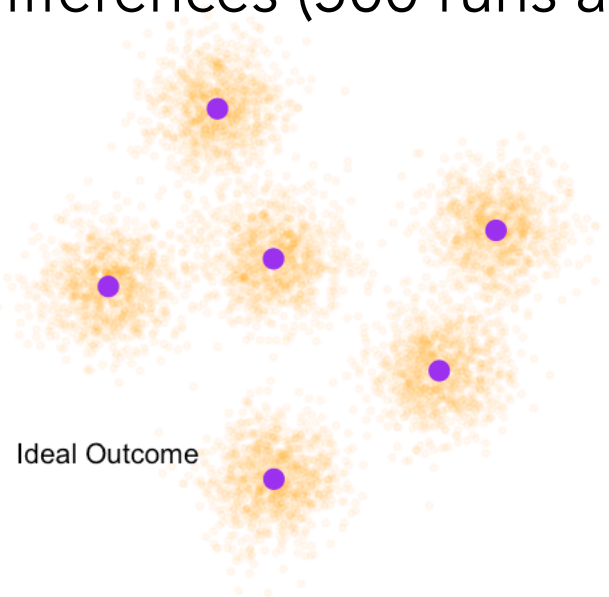
Initialization

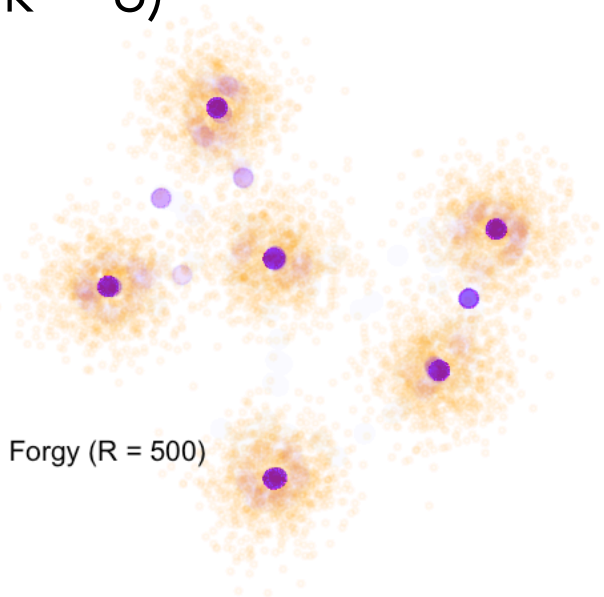
The assignment of the point matters as k-means are deterministic algorithms. Types of initialization:

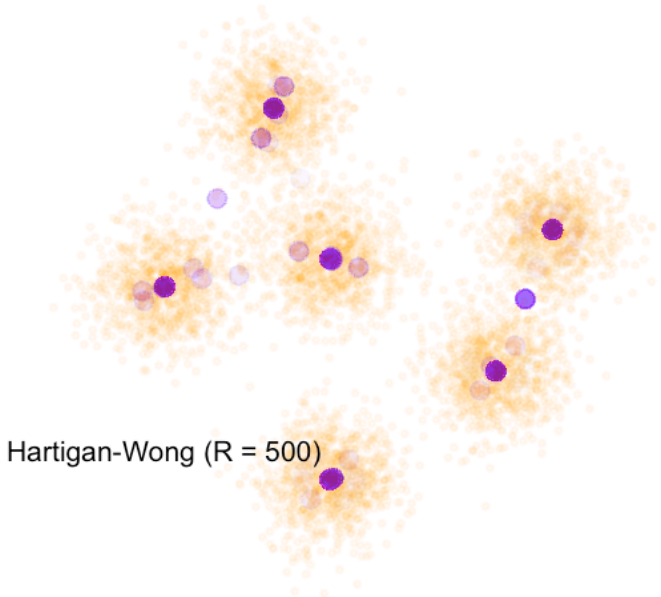
- Forgy. Randomly select k points at once.
- Random Partition. Randomly assign each point to k groups.
- Hartigan Wong: A bit more complicated http://www.labri.fr/perso/bpinaud/userfiles/downloads/hartigan_1979_kmeans.pdf
- Kmeans++. Places points one at a time at positions that are far away from one another to maximize chance that two points are not co-located.

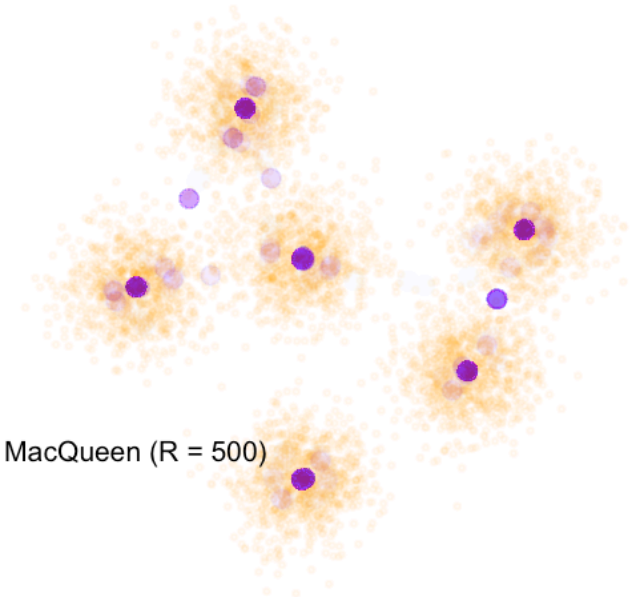
Convergence Differences (500 runs assuming k = 6)







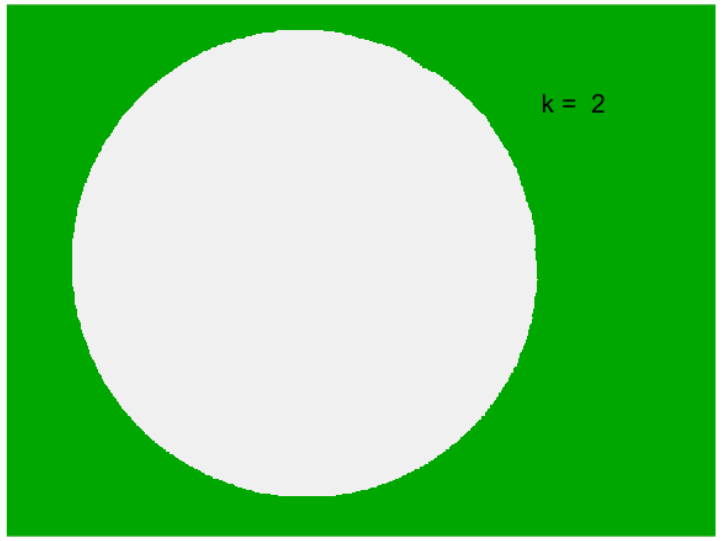


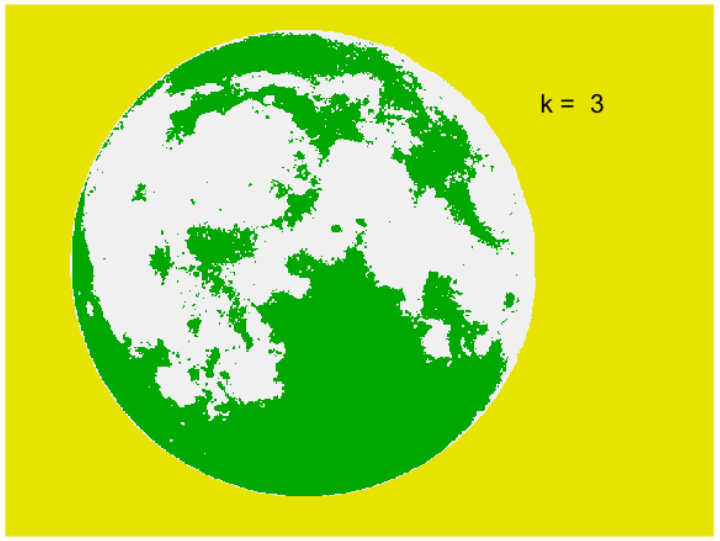


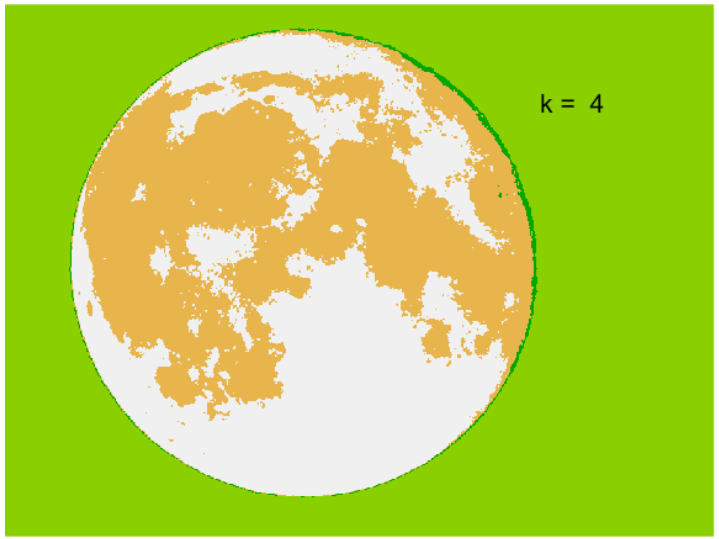
While many runs converged on the right centroid coordinates, a notable amount does not.

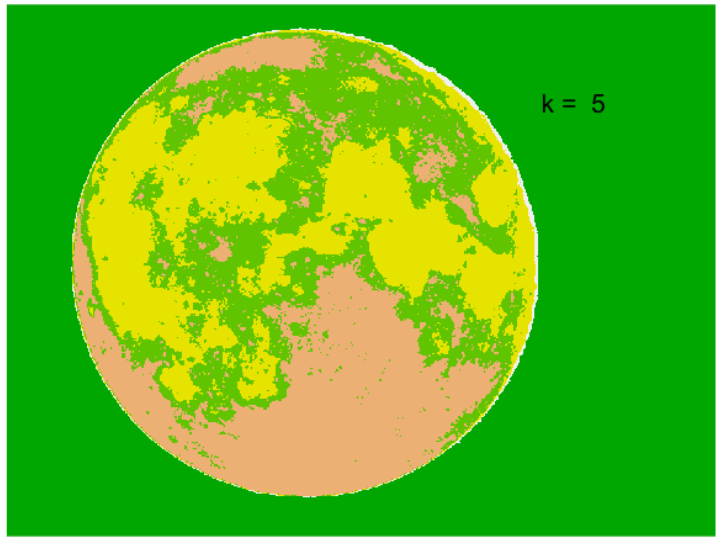
Instant Andy Warhol (k = 2:6)

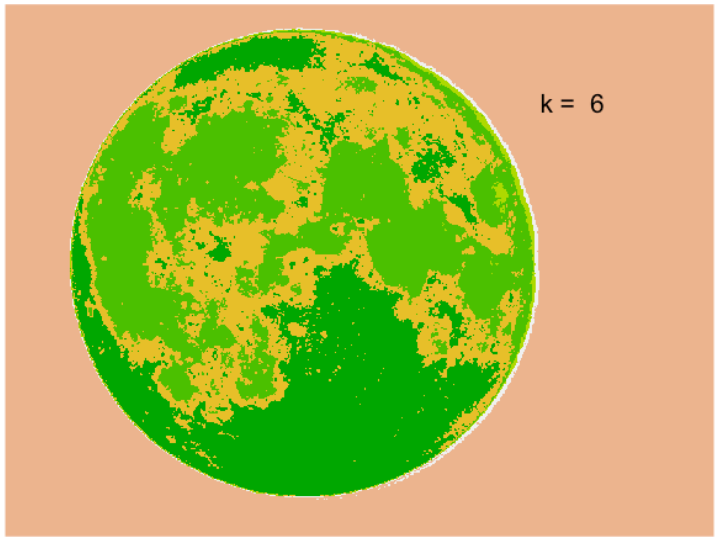
Cutting out the moon (k = 2:6)











Segmenting Markets (k = 4)

Midwestern Company
< $52 million in revenue
<1000 employees
On-prem production
Exporter/Importer

Small Northeastern Company
<$10 milion
<100 employees
Product design
Outsource

Single Owner Operator
>$5 million and < $10 million
Legal services
Logistics focused

Large Businesses
> $1bn
Tech Industry
Non-Exporter

Assumptions

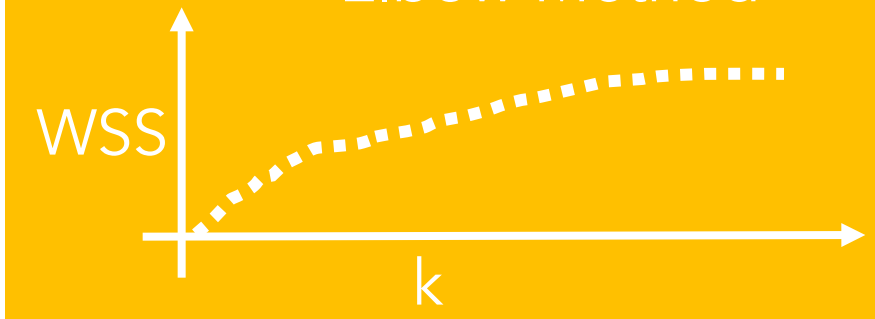
- Can use any distance measure:
  - Euclidean distance: Continuous Real Numbers only.
  - Cosine Similarity: Binary and mixed values

- All features need to be scaled (mean centered/ SD normalized) – all variables have equal weight.

Issues

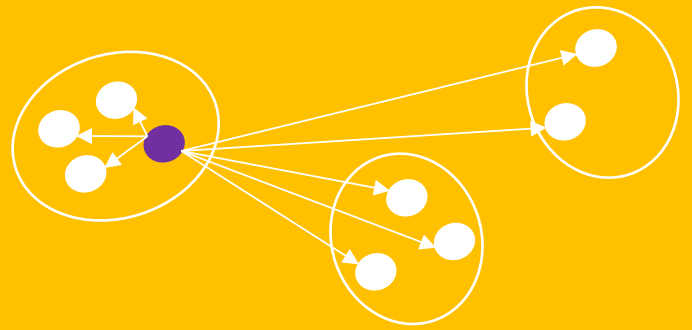
- Which k?
  - Lots of debate in the statistical circles as to how to determine k. Two typical methods include Elbows and Silhouette.

Elbow Method



Find the k where the WSS "bends". WSS does not always bend…

Silhouette Method

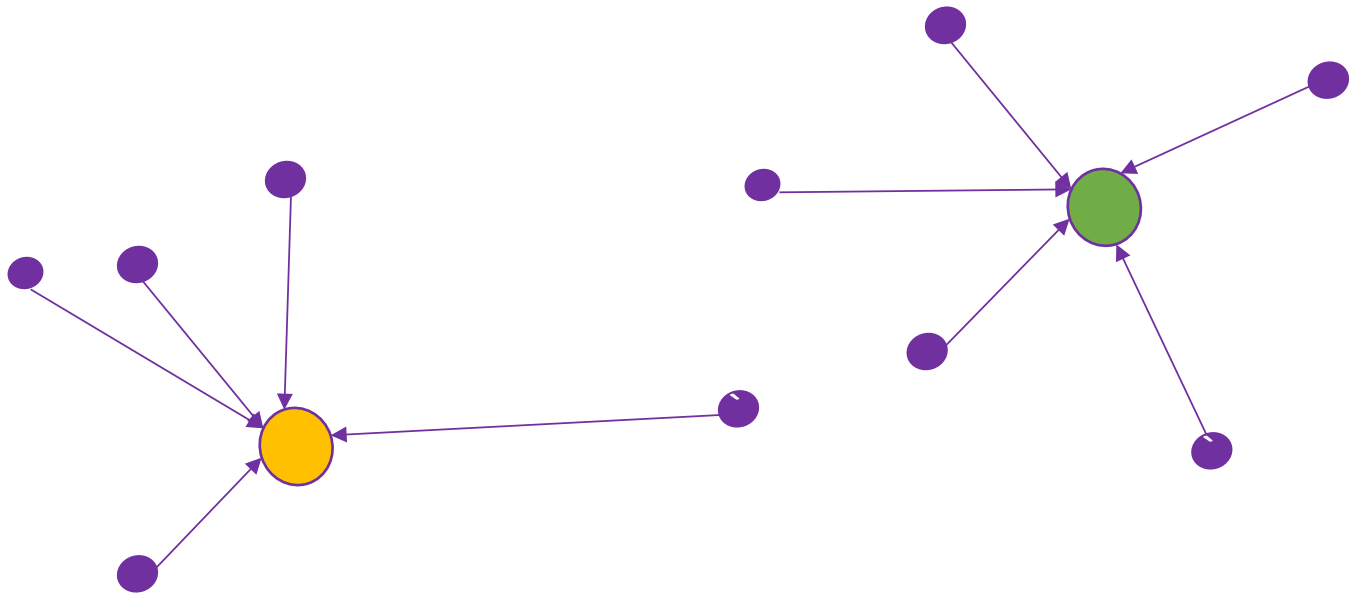


Find k such that the silhouette is maximized (statistic that compares how similar it is to its own cluster vs. others)

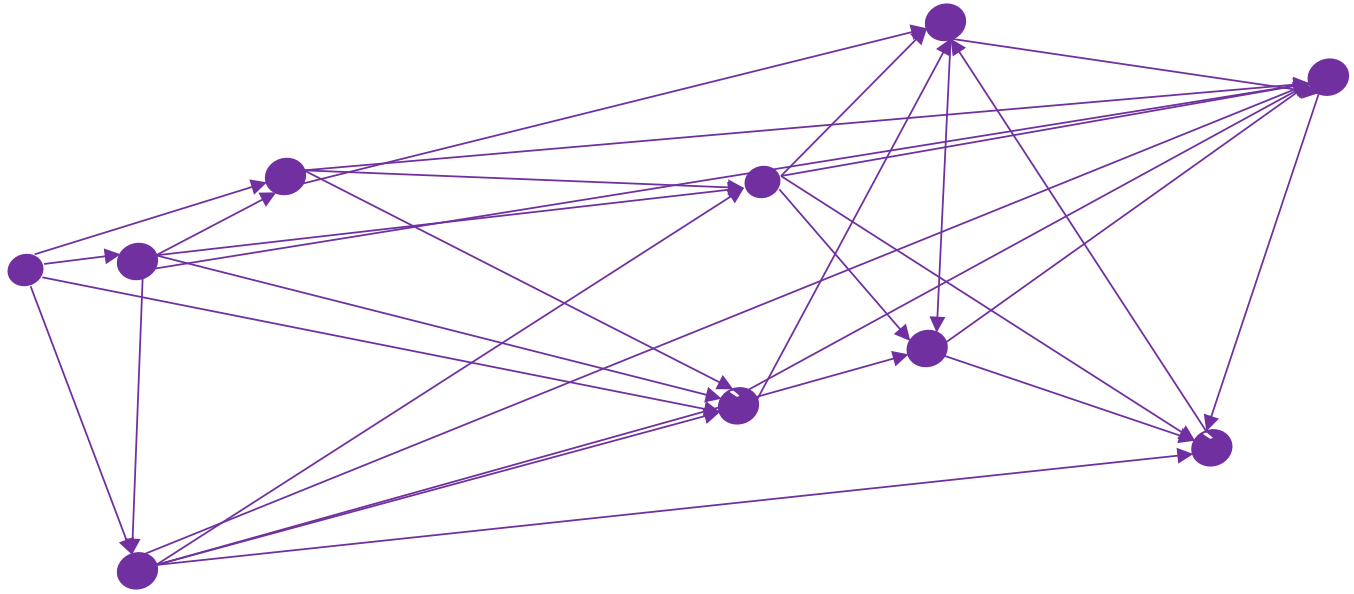
<Code Time/>

Roadmap

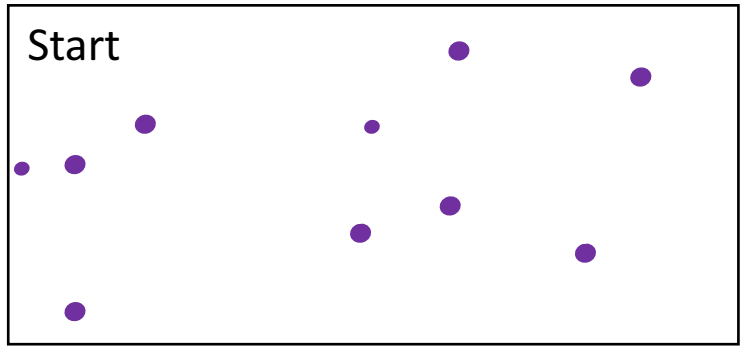
- Last thoughts on classifiers
- Unsupervised learning
- K-Means
- <break>
- Hierarchical clustering

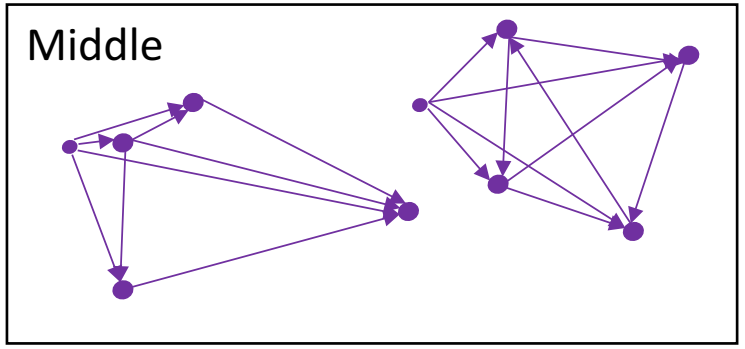


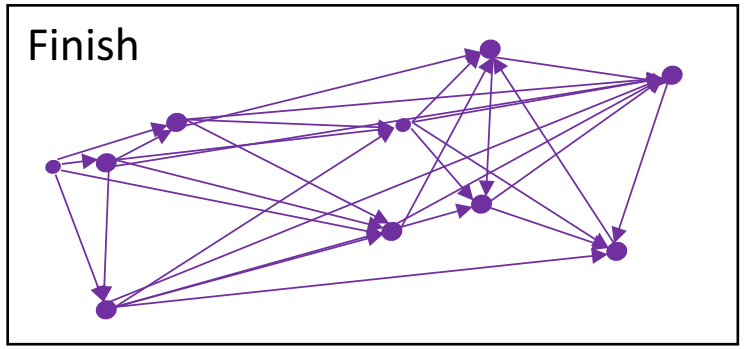
K-means relies on distance to centroids, but does not consider proximity to other similar points.



Agglomerative clustering groups points together based on proximity to one another.



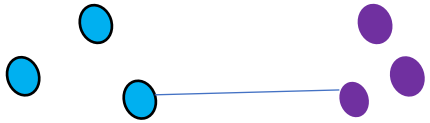
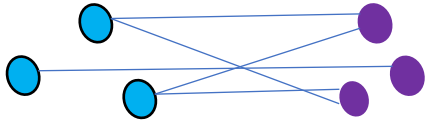
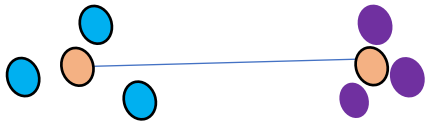




An algorithm that is a bit more computationally intensive and less arbitrary.

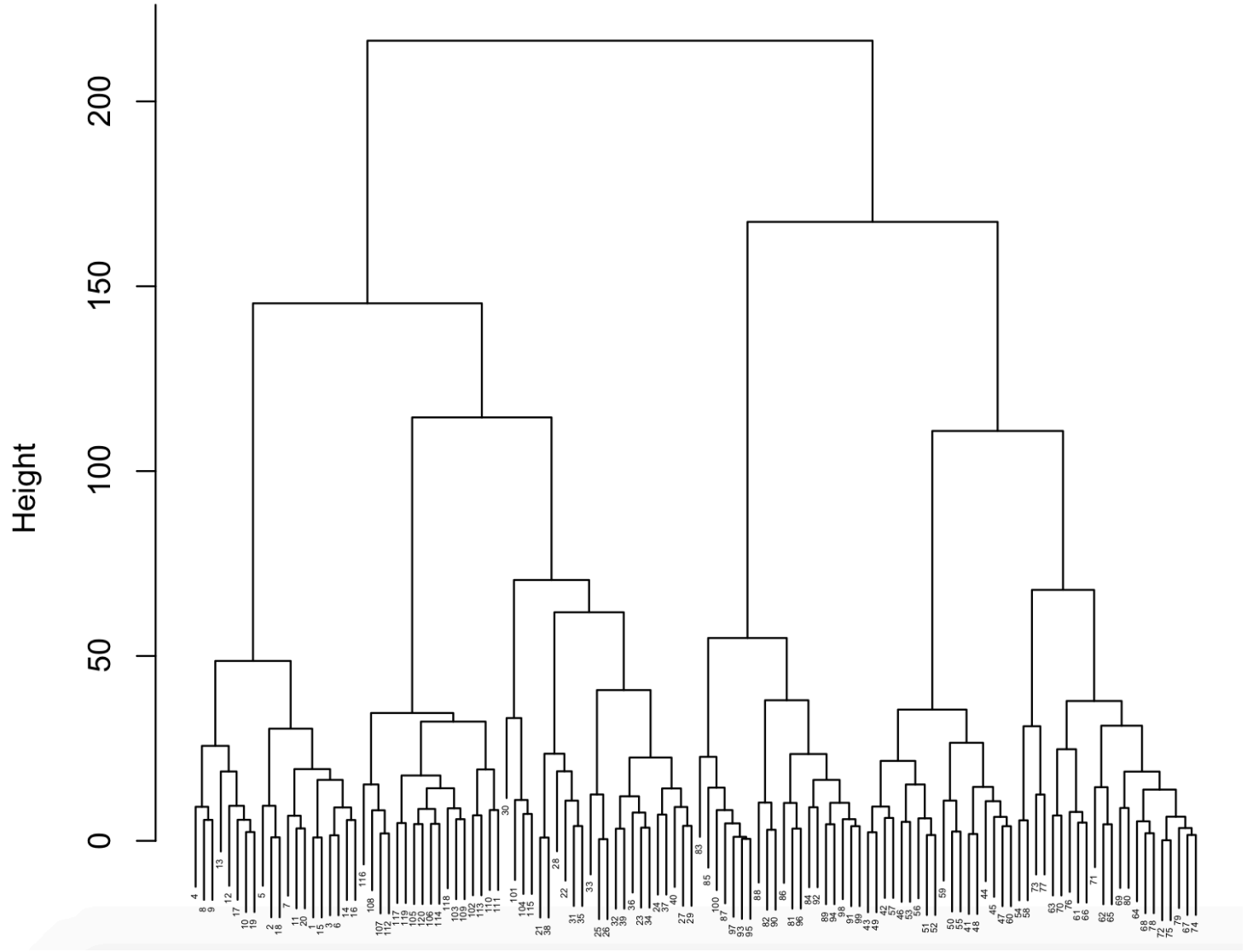
```
Calculate distance d between all points
To start, all points are set as their own clusters (singletons)
Until there is only one cluster:
    Find the closest pair of clusters in terms of linkage distance
    Merge into a single cluster
    Recalculate distances from new cluster to all other clusters
Stop when all points are in one cluster
```

Many different ways of measuring distance

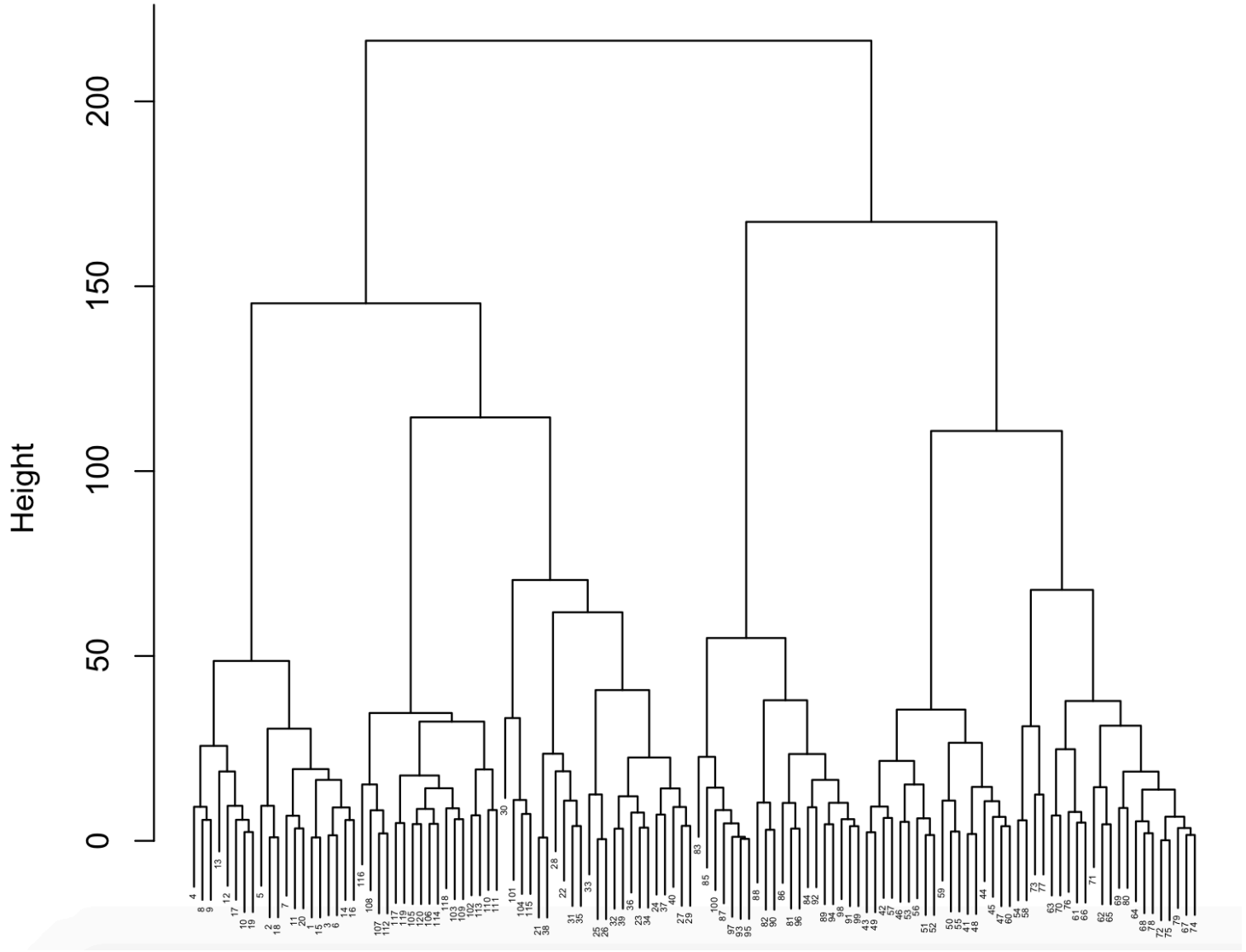
Linkage distance	Formula	Choose clusters based on:	Graphic
Single Linkage	$\min(d(x_i, y_j))$	distance between the two closest points in two clusters	
Complete Linkage	$max(d(x_i, y_j))$	Distance between the two farthest points in two clusters	
Average Linkage	$\sum_{i}^{k}\sum_{j}^{l} d(x_i, y_j)$	Average distance between all points	
Centroid Distance	$max(d(\bar{x}, \bar{y}))$	Distance between cluster centroids	
Ward's Distance	$\frac{\lVert \overline{x_k} - \bar{x_l} \rVert^2}{\frac{1}{N_k}\frac{1}{N_l}}$	Clusters that minimize variance increase by using an ANOVA sum of squares overall all partitions	

Dendrograms represent all the linkages.

Higher up the chart = greater distance between clusters.



How do we choose clusters?



<Code Time/>