Unsupervised Learning

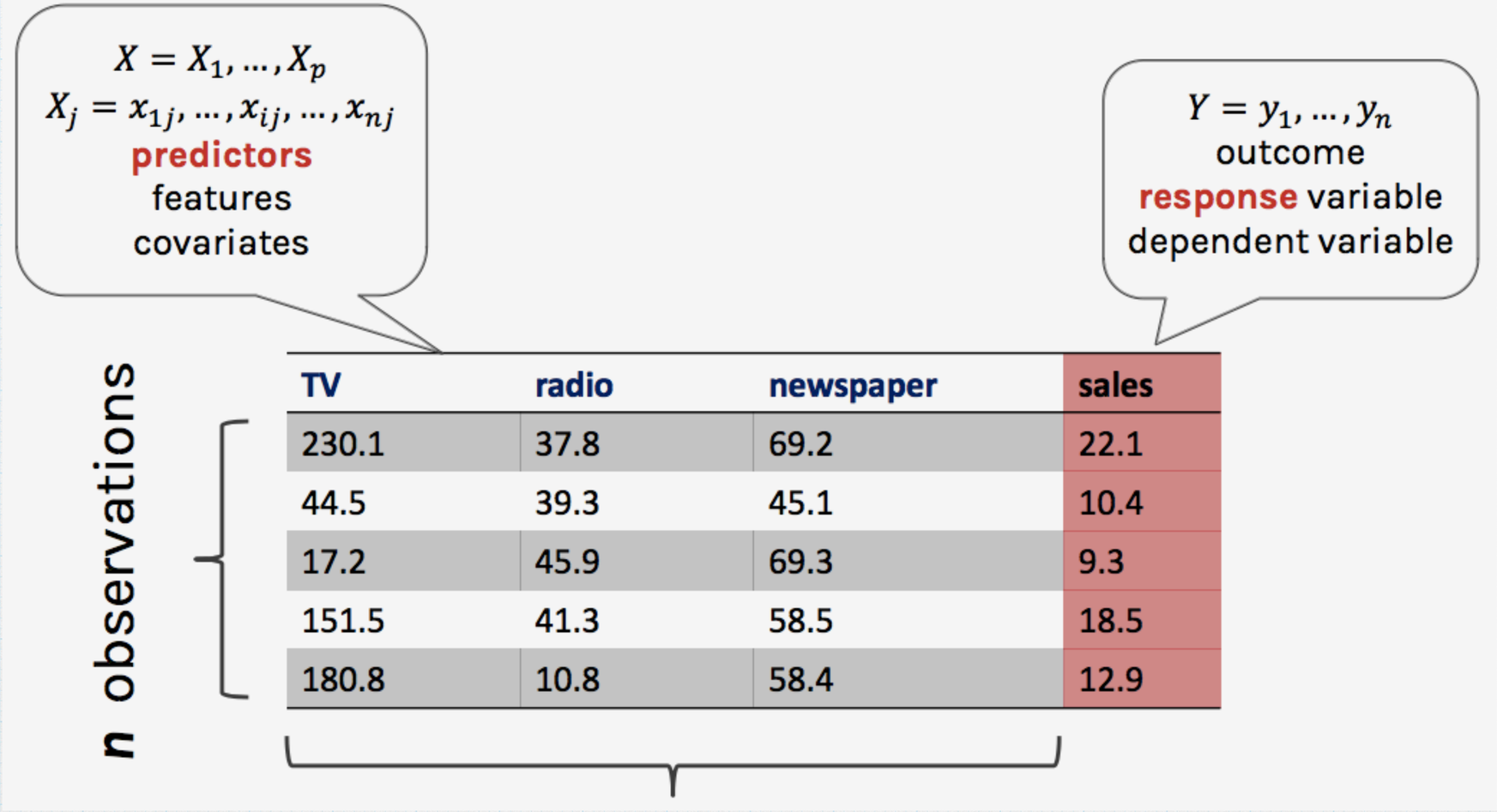
Topics we will cover

* Why unsupervised data?
* Goal 1: Clustering. Goal 2: Identifying structure
* Clustering: k-Means algorithm
* Code block A: k-Means algorithm
* Code block B: k-Means for Image segmentation
* Structure: Principal component analysis
* Code block C: PCA for document analysis
* Future: Representation learning for downstream tasks

A Dataset ...

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

This is supervised



But most are unsupervised!

TV	radio	newspaper	[redacted]
230.1	37.8	69.2	
44.5	39.3	45.1	
17.2	45.9	69.3	
151.5	41.3	58.5	
180.8	10.8	58.4	

No labels? What do we do with it?

Unsupervised

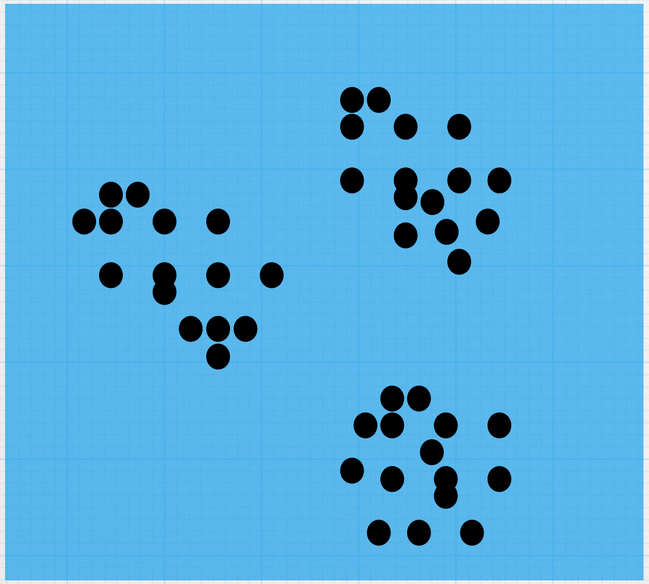
* No labels on data
* Why? That is what we have; labelling is expensive.
* Example:
    * Trillions of images on Google. Labeled? Tiny
    * Youtube videos, genome sequences, Wikipedia
* What do we do with it?

Unsupervised data analysis

* Examples to keep in mind: Text documents, user profiles, credit card transactions, images, videos,
* Machine learning goals?
  * Group similar items together
  * Identify outliers
  * Compress/de-noise
  * Preprocess for future applications

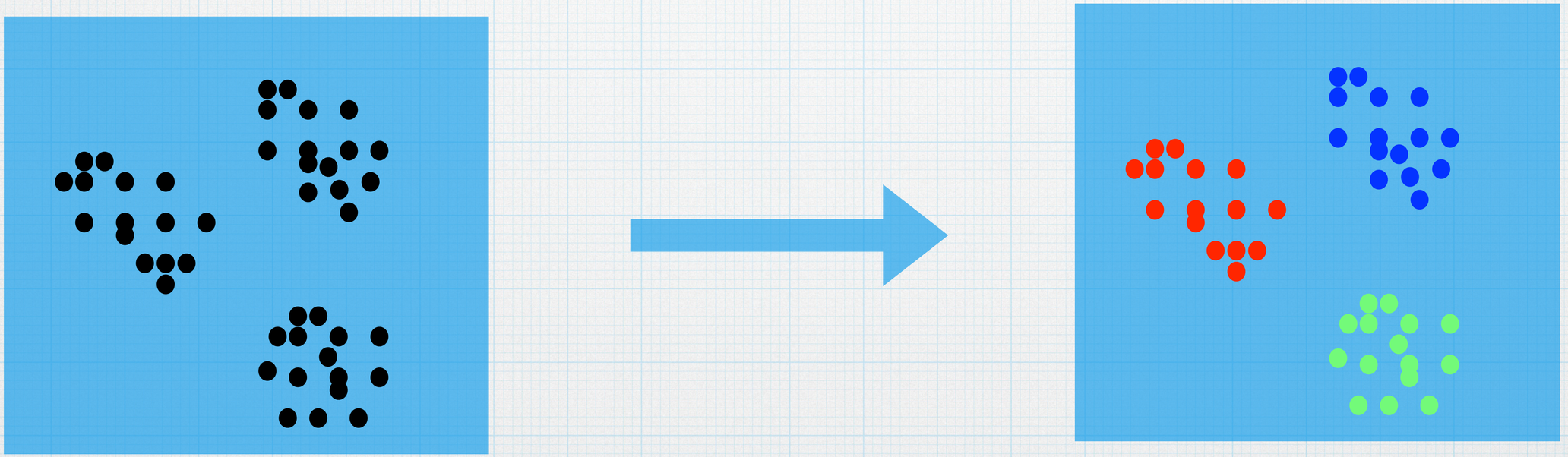
Clustering

* Goal: Given a dataset X, group similar items into groups



Clustering

* Goal: Given a dataset X, group similar items into groups



Clustering

* Goal: Given a dataset X, group similar items into groups
* Marketing and Online ads: Group user profiles and show similar ads
* Recommendations: Group netflix viewer ratings and suggest new ratings
* Fraud detection: Group credit card transactions
* Fake news/Spam filters: Group texts
* Creating groups on social networks

What is a good clustering?

* Goal: Given a dataset X, group similar items into groups

* Property 1: Elements in each cluster be "close"
* Property 2: Across group distances large
* Ideal: Use as few clusters as possible

How to measure distance?

Let us define the proble mathematically ...

Clustering Algorithms

Each row is an example

x1	0.1	0.2	0.3	0.4
x2	1	0.1	-0.1	-0.5
	4	3	1	0
	-3	1	-2	1
xn	0	0	0.1	0.2

Each column is a feature

Clustering Algorithms

Each row is an example

x1	0.1	0.2	0.3	0.4	→ Group 1
x2	1	0.1	-0.1	-0.5	→ Group 2
	4	3	1	0	→ Group 2
	-3	1	-2	1	→ Group 3
xn	0	0	0.1	0.2	→ Group 1

Each column is a feature

Clustering Algorithm

Clustering Algorithms

Each row is an example

x1	0.1	0.2	0.3	0.4
x2	1	0.1	-0.1	-0.5
	4	3	1	0
	-3	1	-2	1
xn	0	0	0.1	0.2

Each column is a feature

Dataset $X = \{x_1, x_2, \ldots, x_n\}$
Each point x has d coordinates

Clustering Algorithm:
Input: X, num. clusters k
Output: Grouping ...

$$C(x_1) = 2$$
$$C(x_1) = 3$$
$$\ldots$$
$$C(x_n) = 1$$

What is a good clustering?

Ideal: Within cluster points
are close to each other

How distance is measured is critical!
One option: Distance between them in space.

Dataset $X = \{x_1, x_2, \ldots, x_n\}$
Each point x has d coordinates

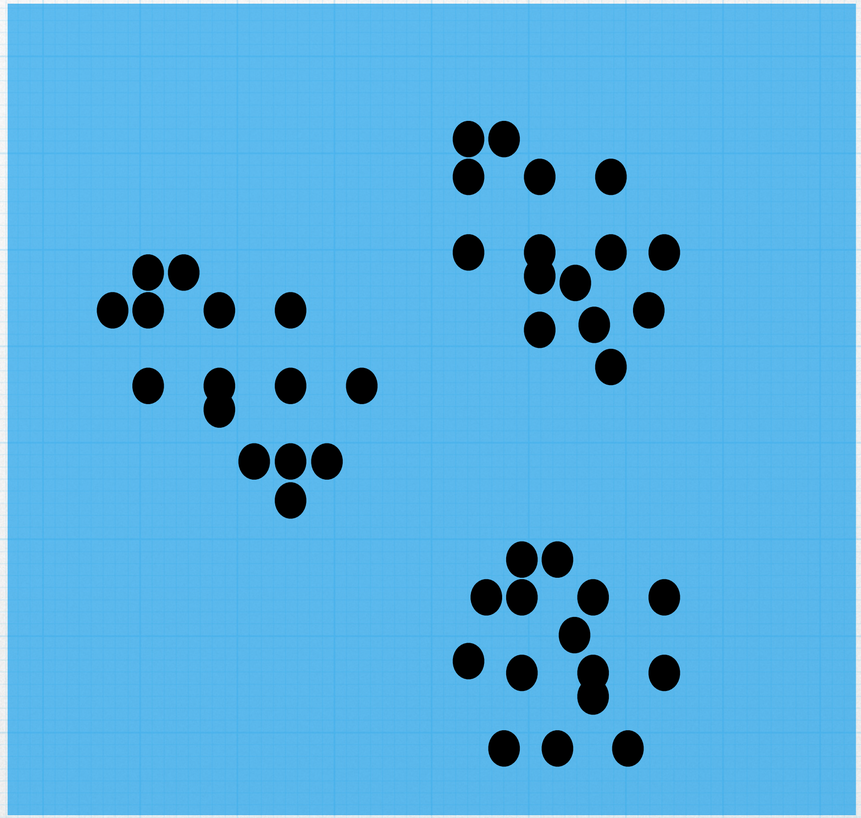
Clustering Algorithm:
Input: X, num. clusters k
Output: Grouping ...

$$C(x_1) = 2$$
$$C(x_1) = 3$$
$$\ldots$$
$$C(x_n) = 1$$

k-Means Algorithm

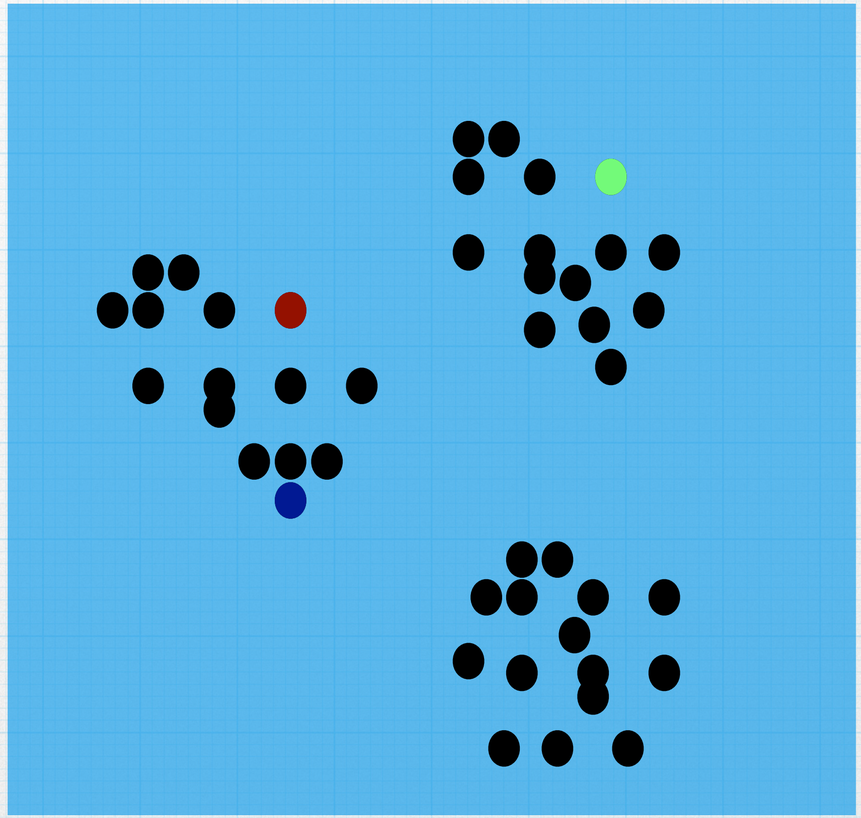
k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - Assign data points to closest center
  - Move center to average of points in cluster
- Stop when no changes



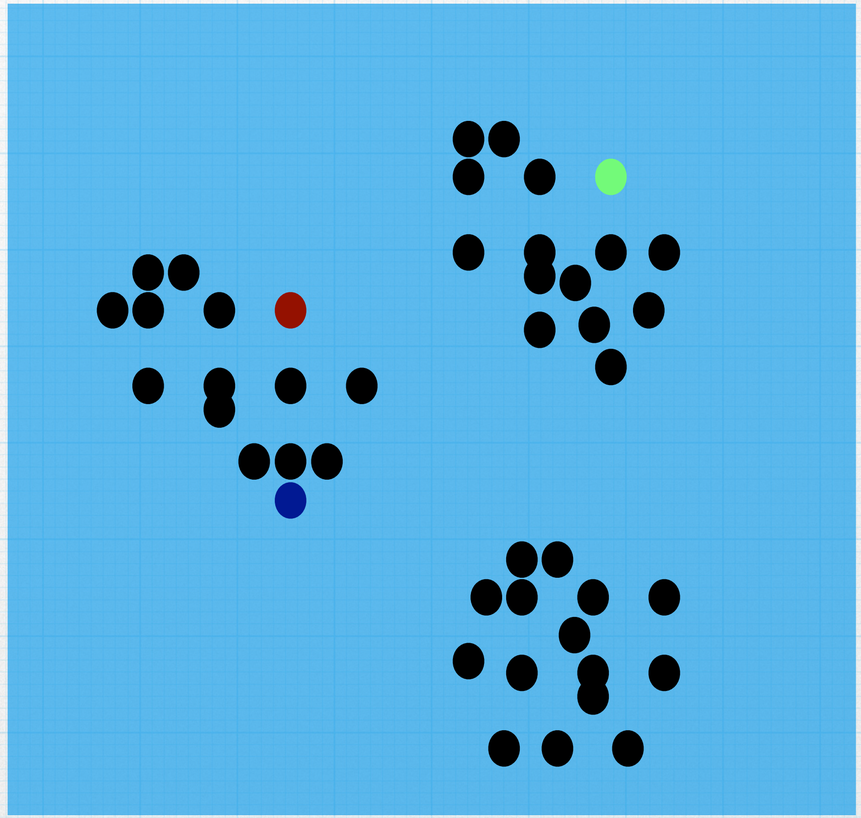
k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - Assign data points to closest center
  - Move center to average of points in cluster
- Stop when no changes



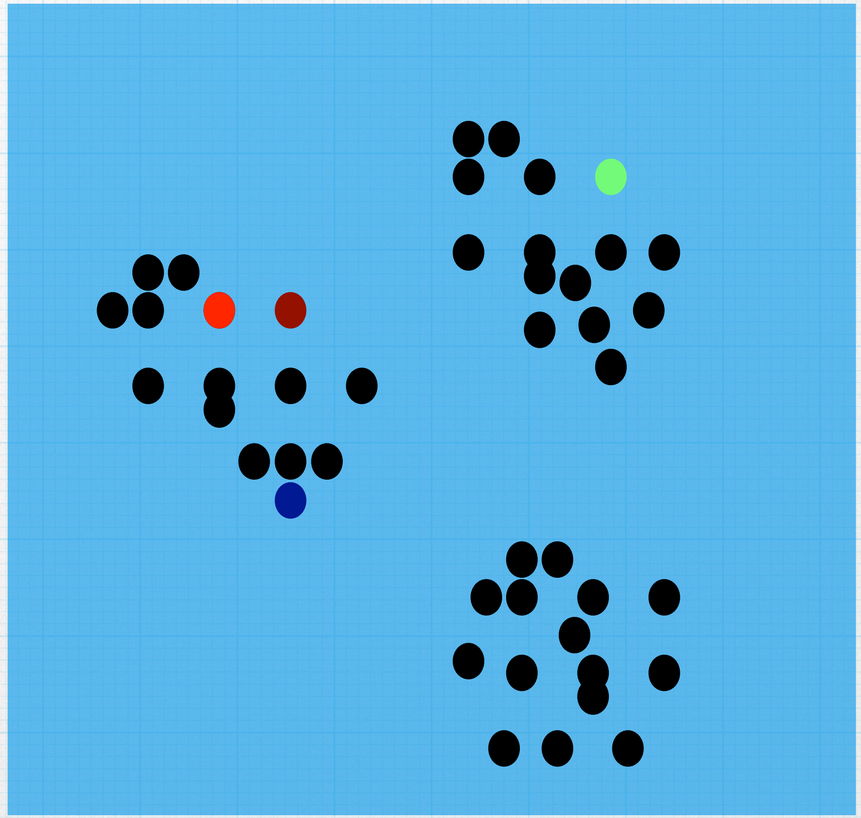
k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - **Assign data points to closest center**
  - Move center to average of points in cluster
- Stop when no changes



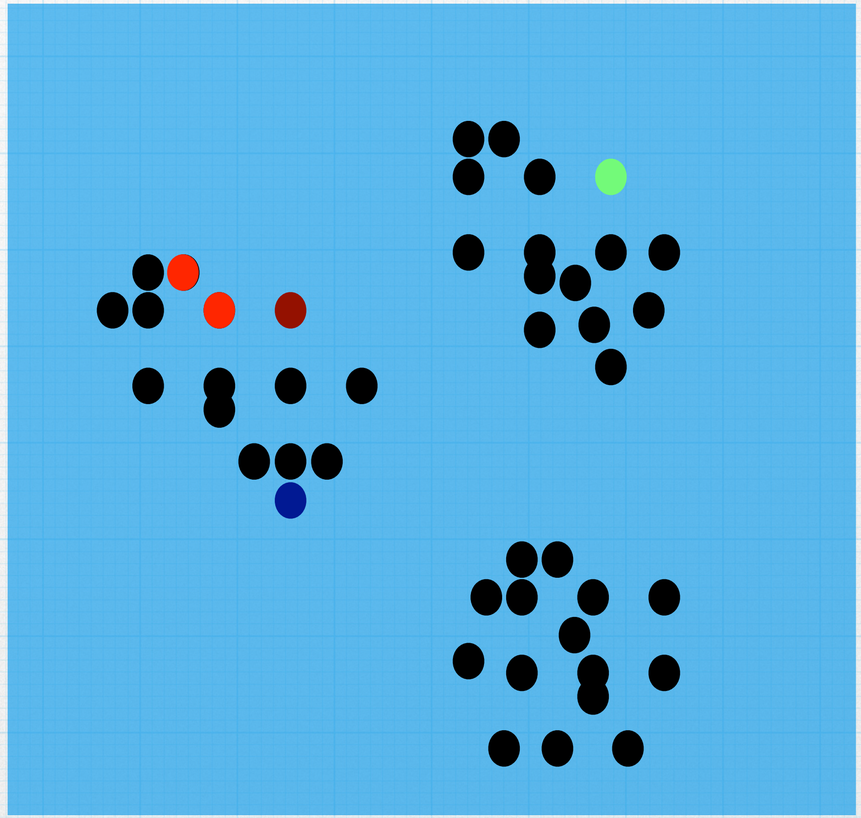
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



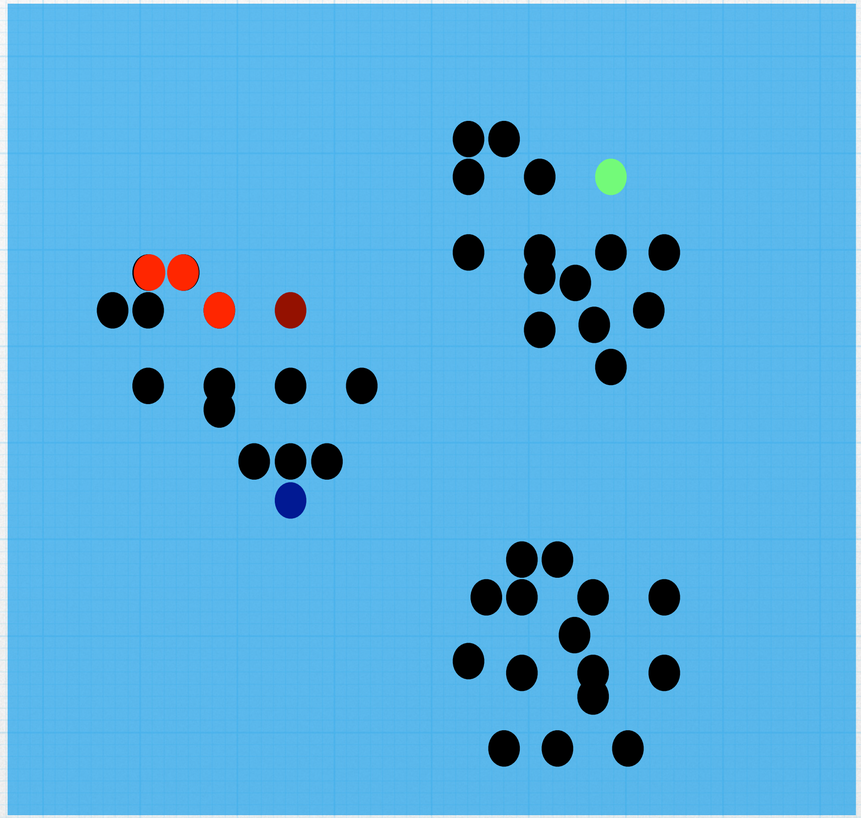
k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - **Assign data points to closest center**
  - Move center to average of points in cluster
- Stop when no changes



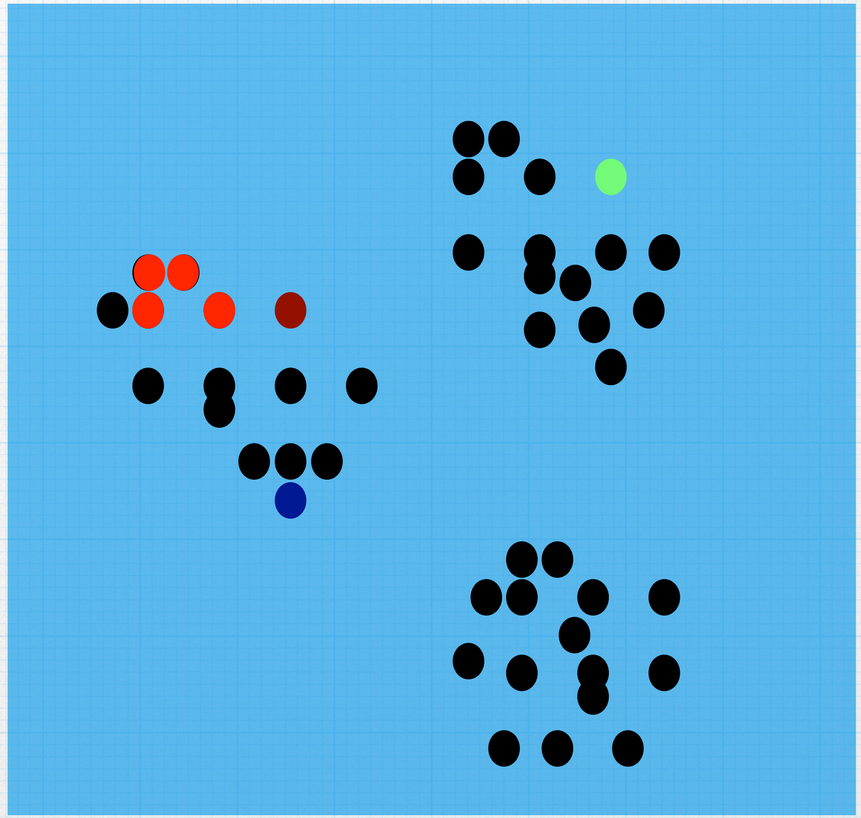
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



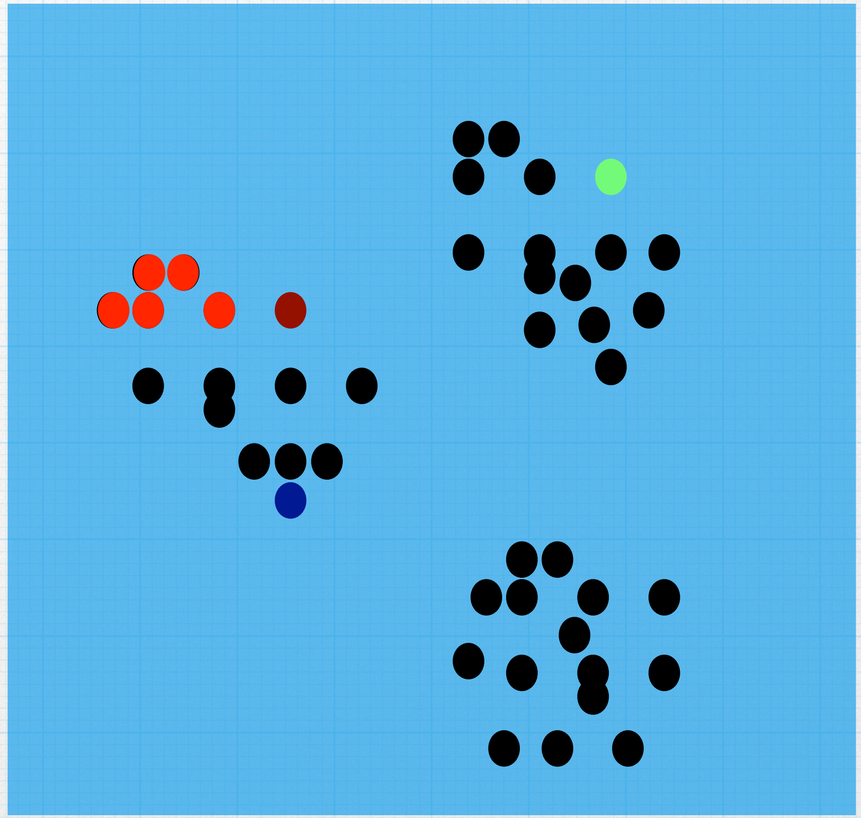
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



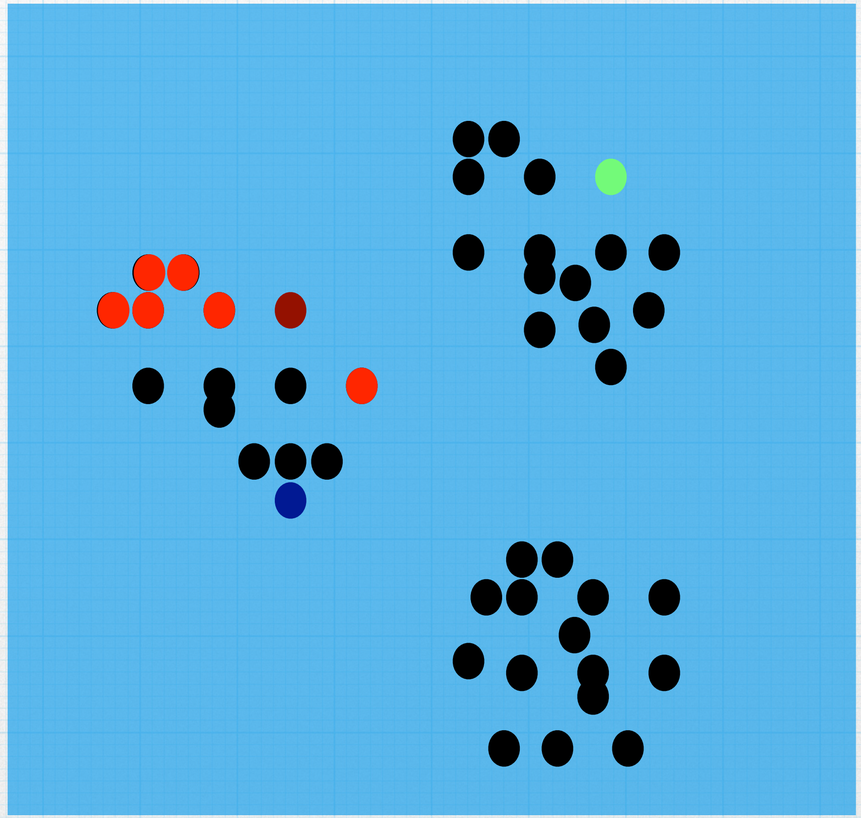
k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - **Assign data points to closest center**
  - Move center to average of points in cluster
- Stop when no changes



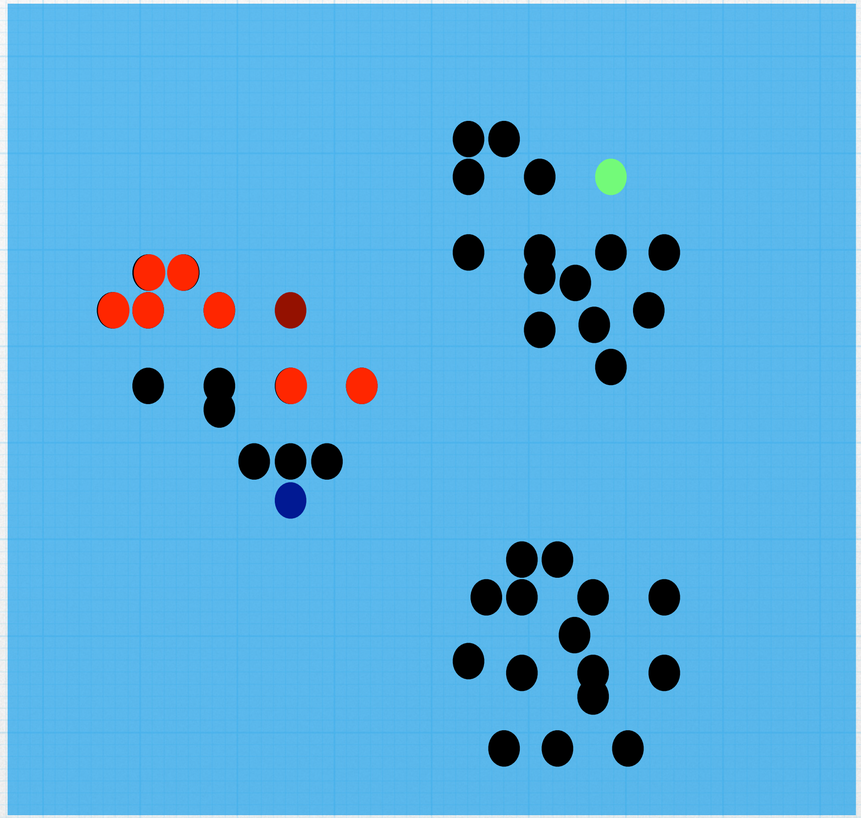
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



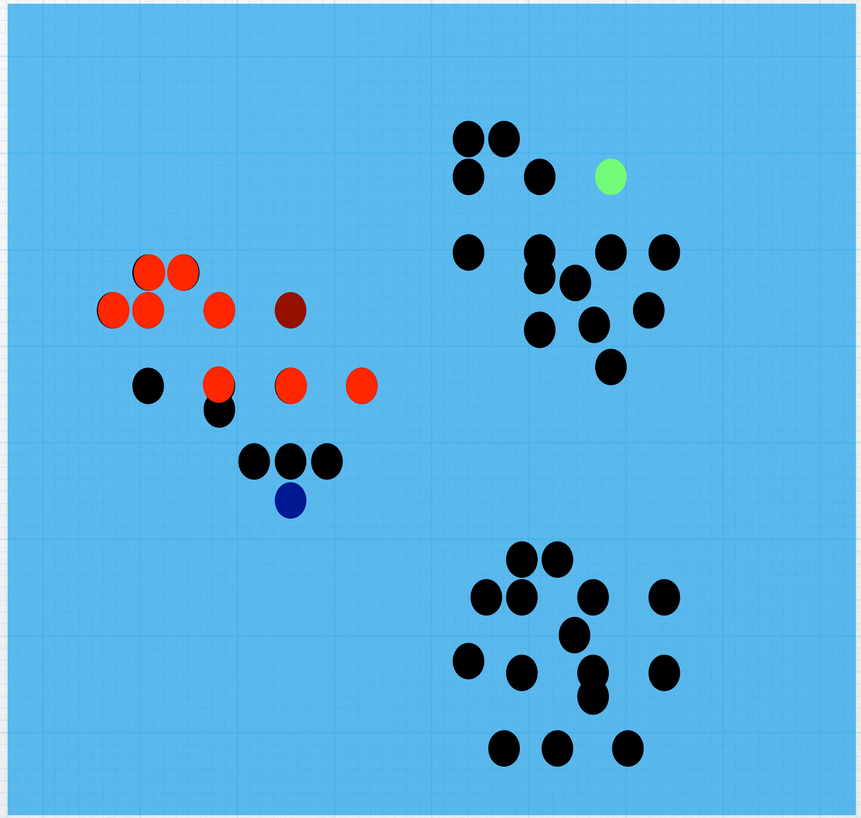
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



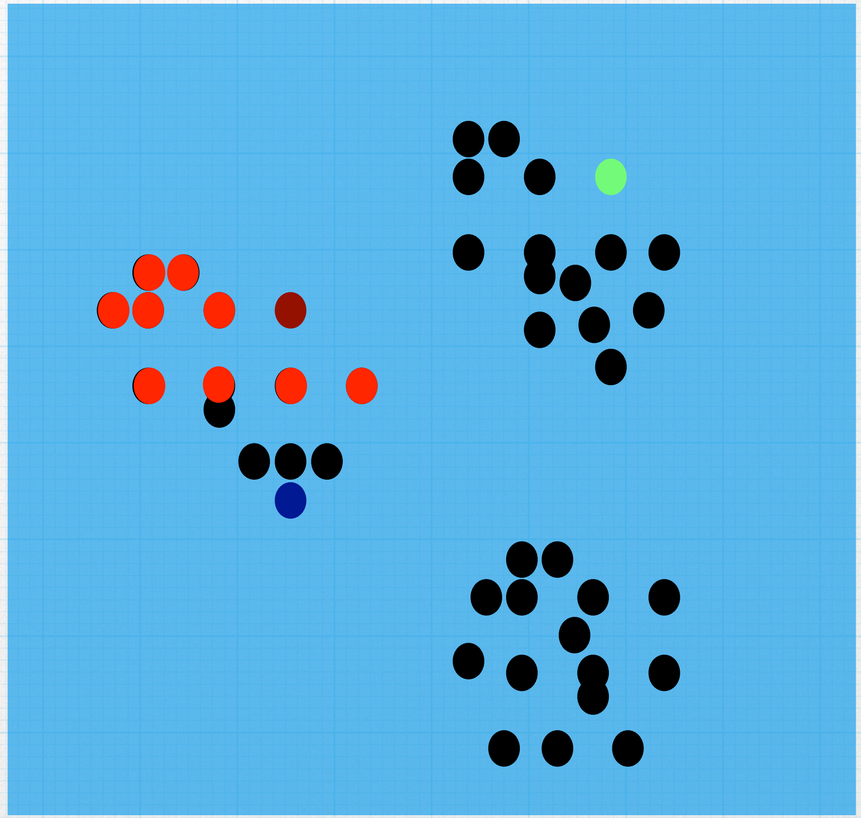
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



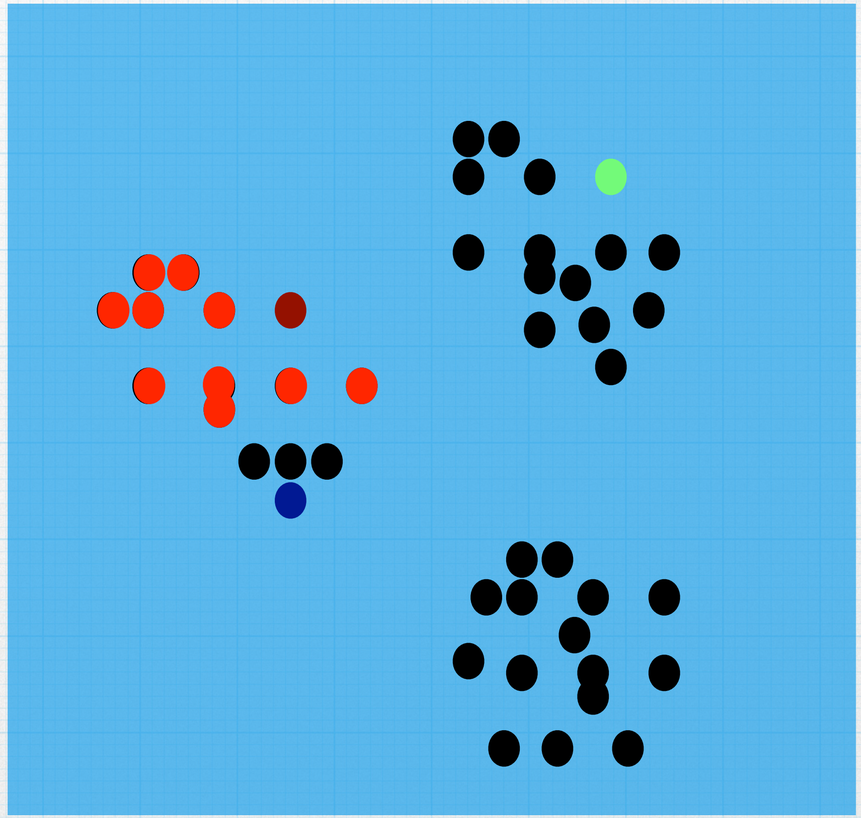
k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - **Assign data points to closest center**
  - Move center to average of points in cluster
- Stop when no changes



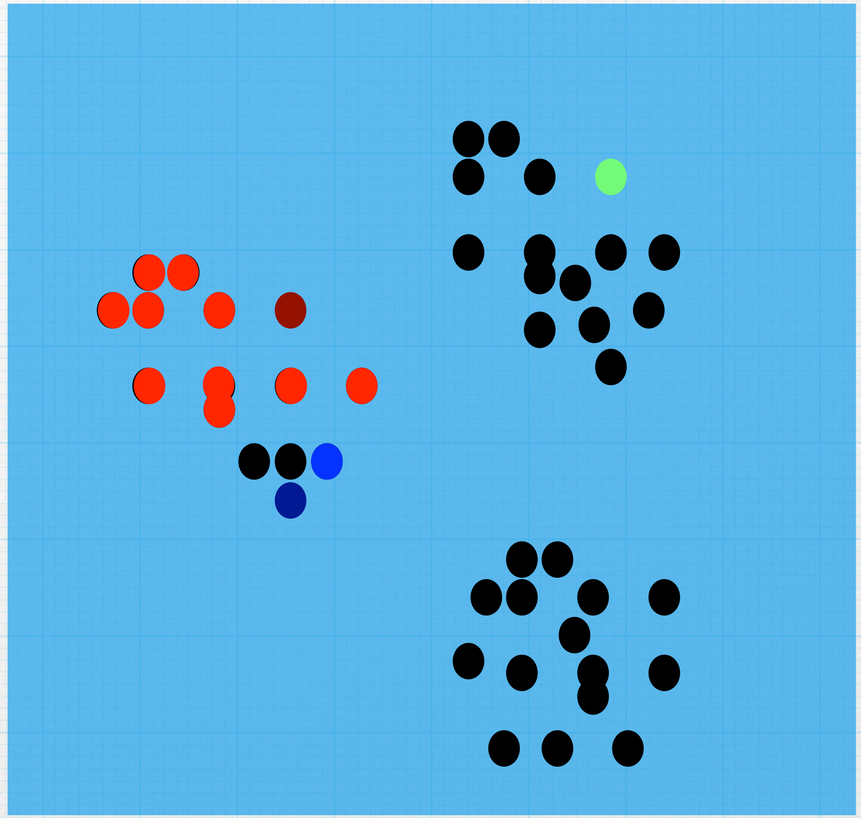
k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - **Assign data points to closest center**
  - Move center to average of points in cluster
- Stop when no changes



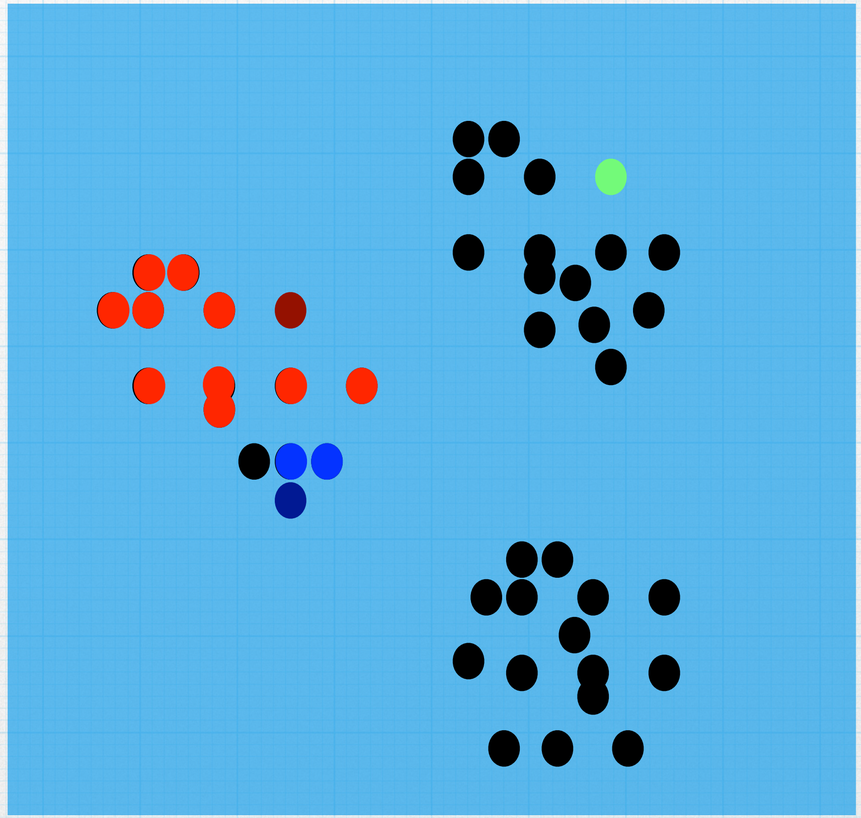
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



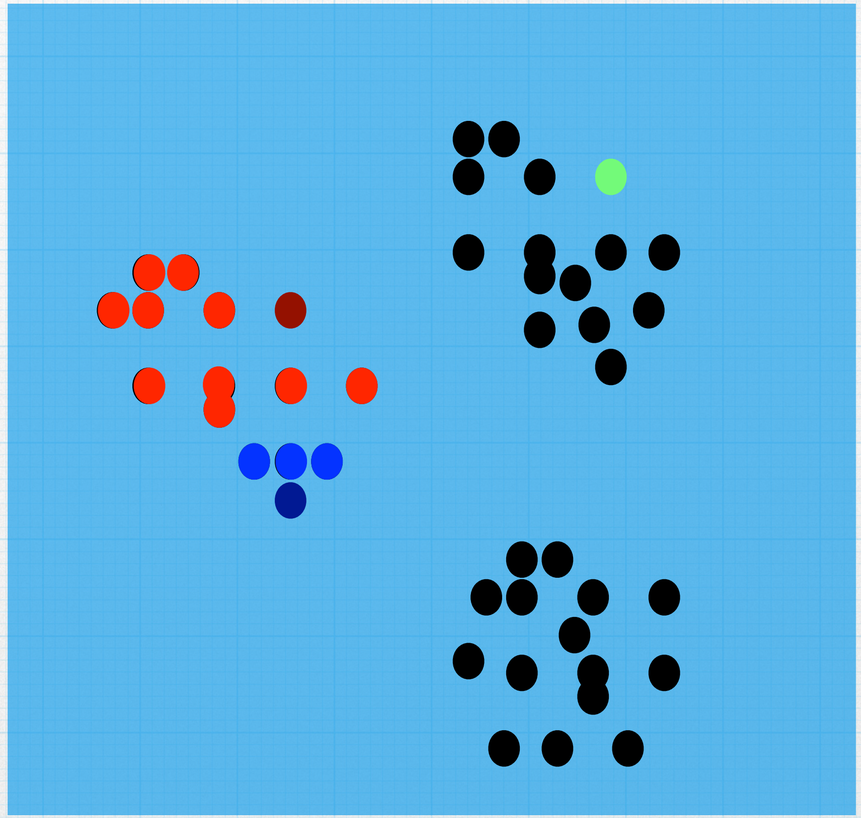
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



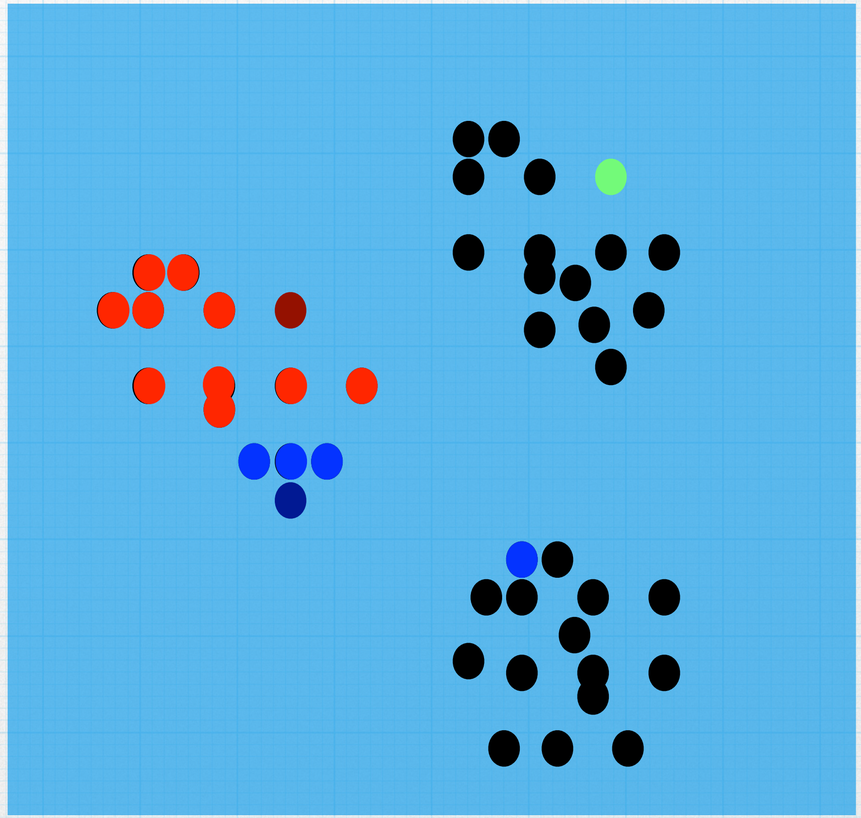
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



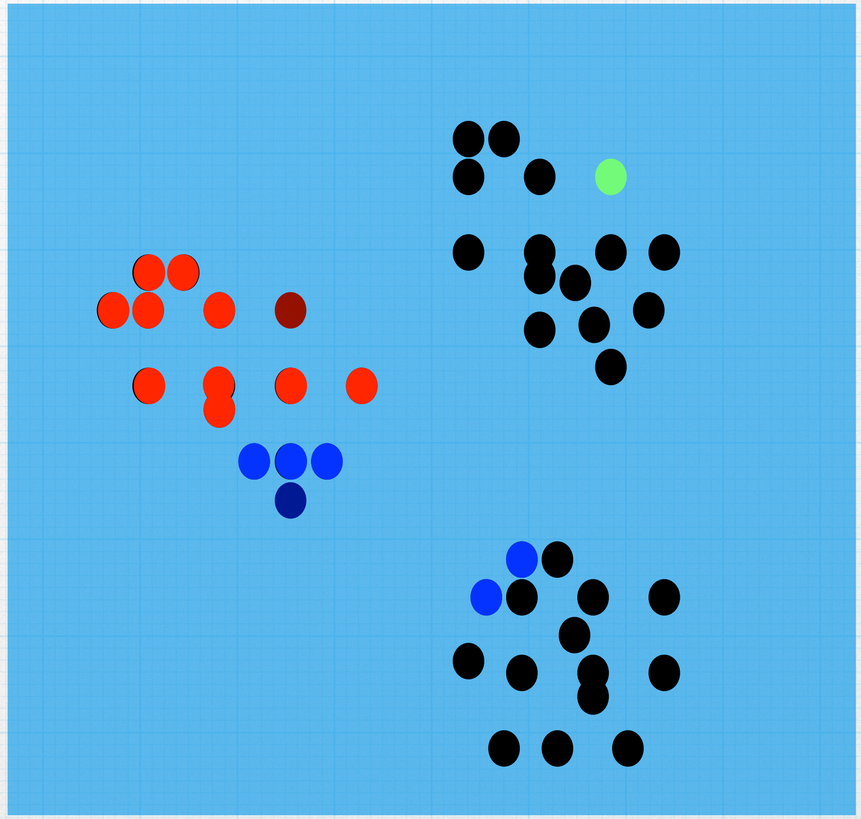
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



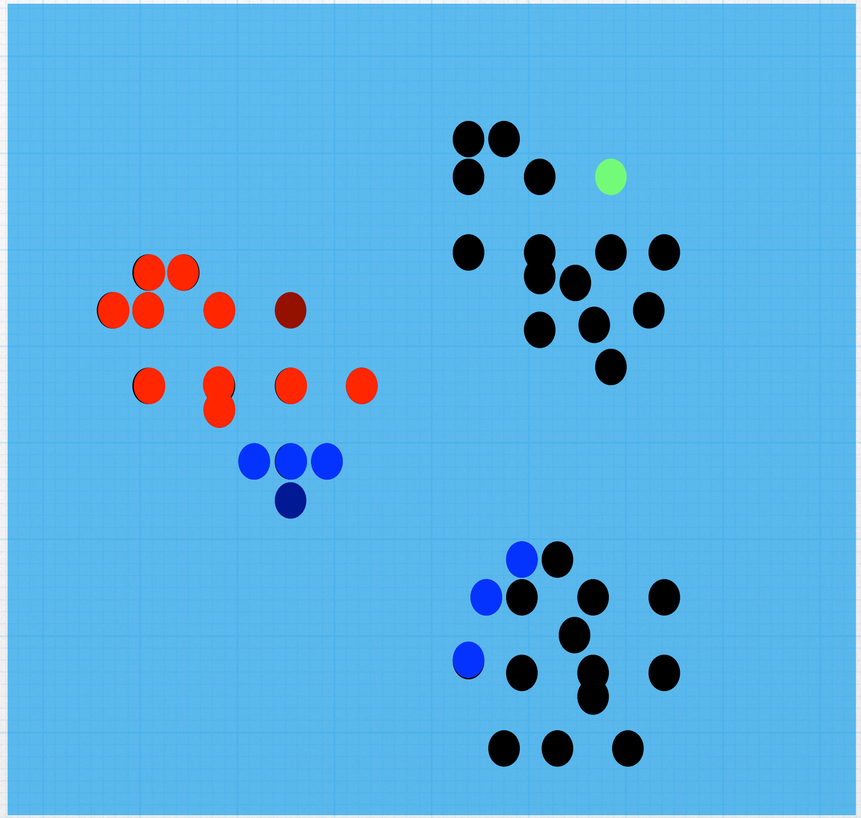
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



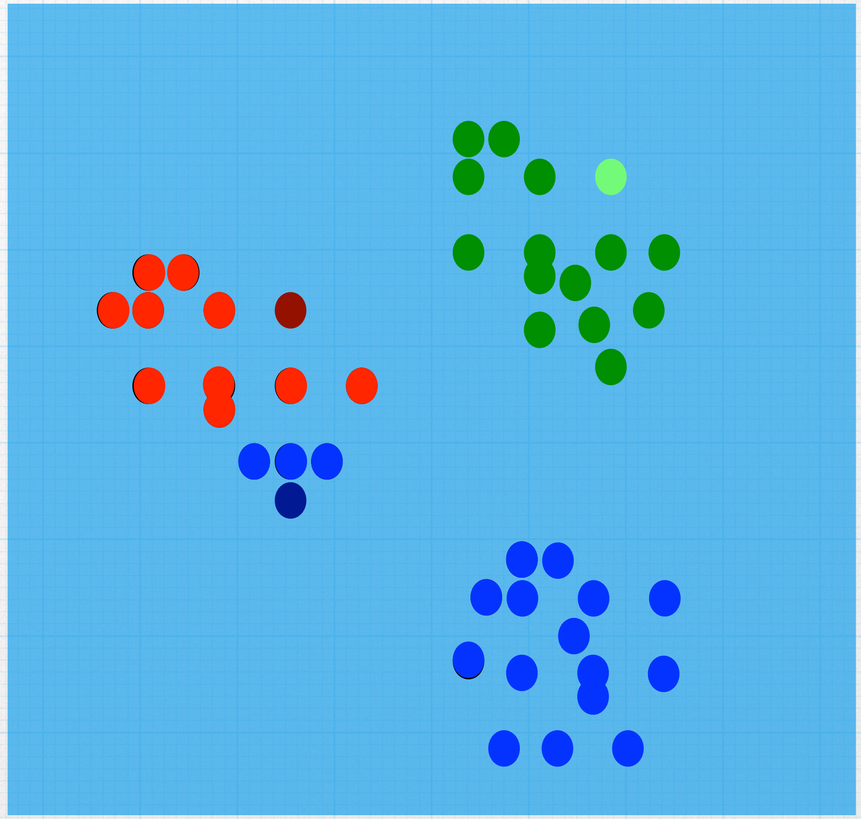
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



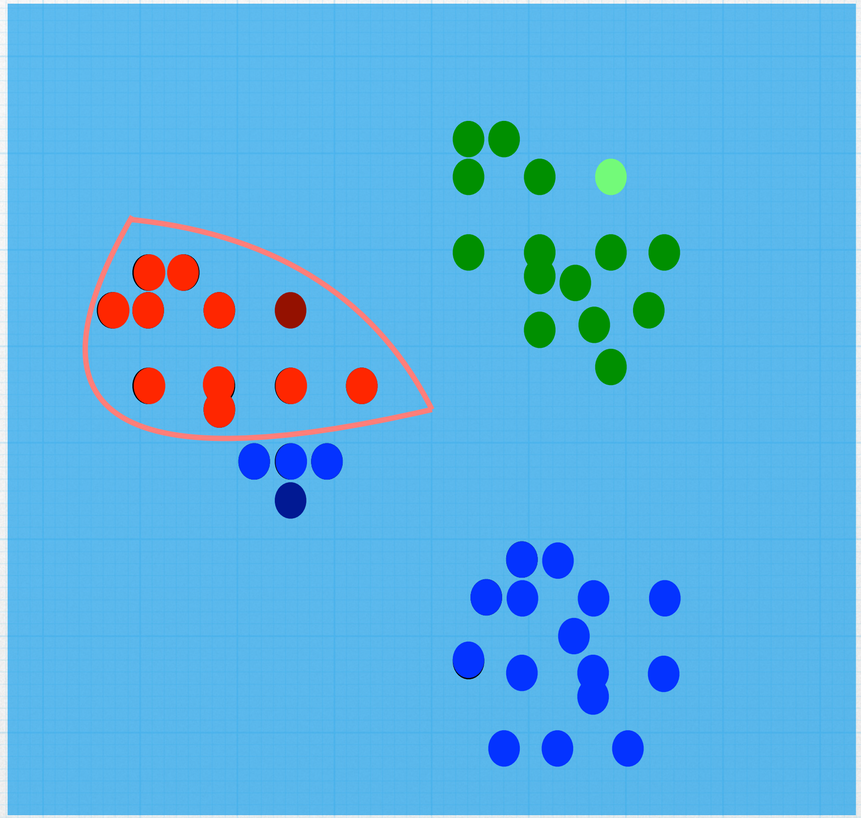
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * **Assign data points to closest center**
  * Move center to average of points in cluster
* Stop when no changes



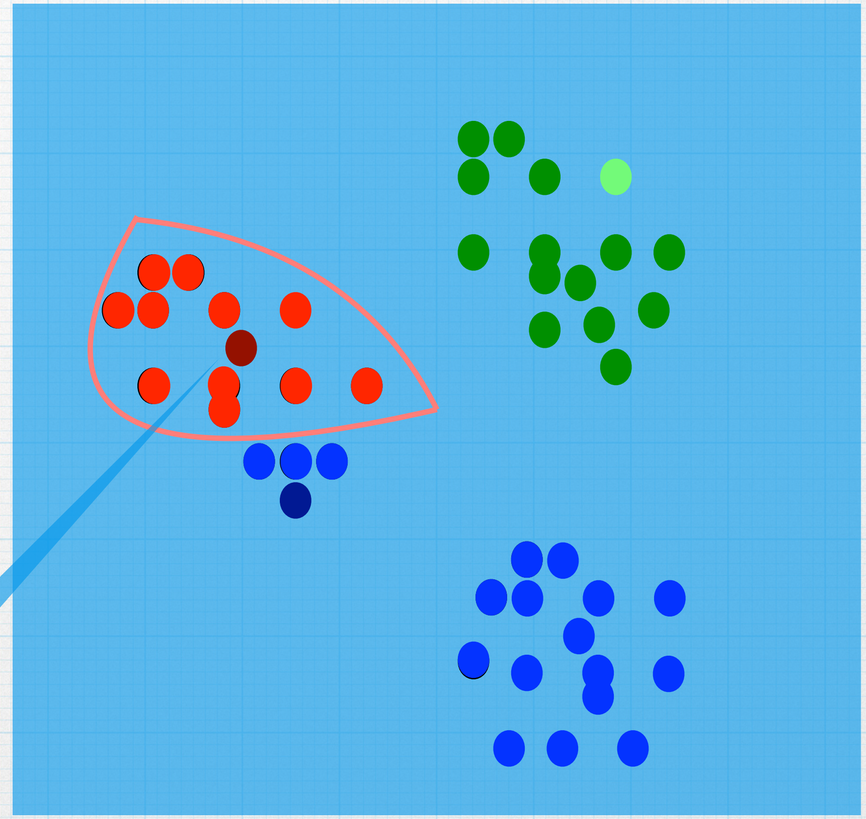
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * Assign data points to closest center
  * **Move center to average of points in cluster**
* Stop when no changes



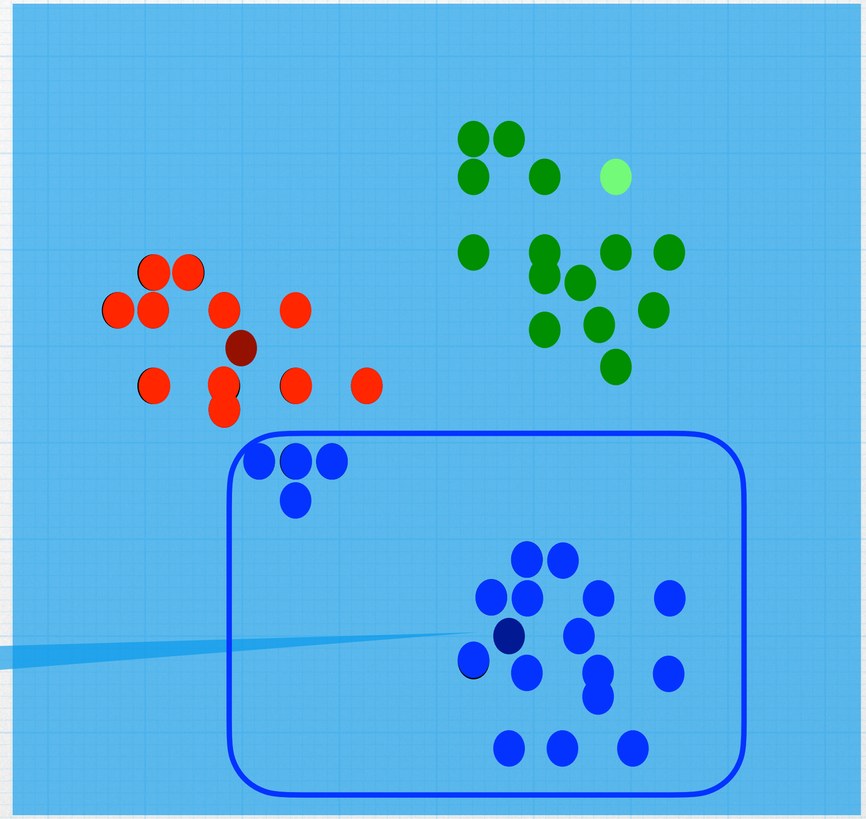
k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - Assign data points to closest center
  - **Move center to average of points in cluster**
- Stop when no changes



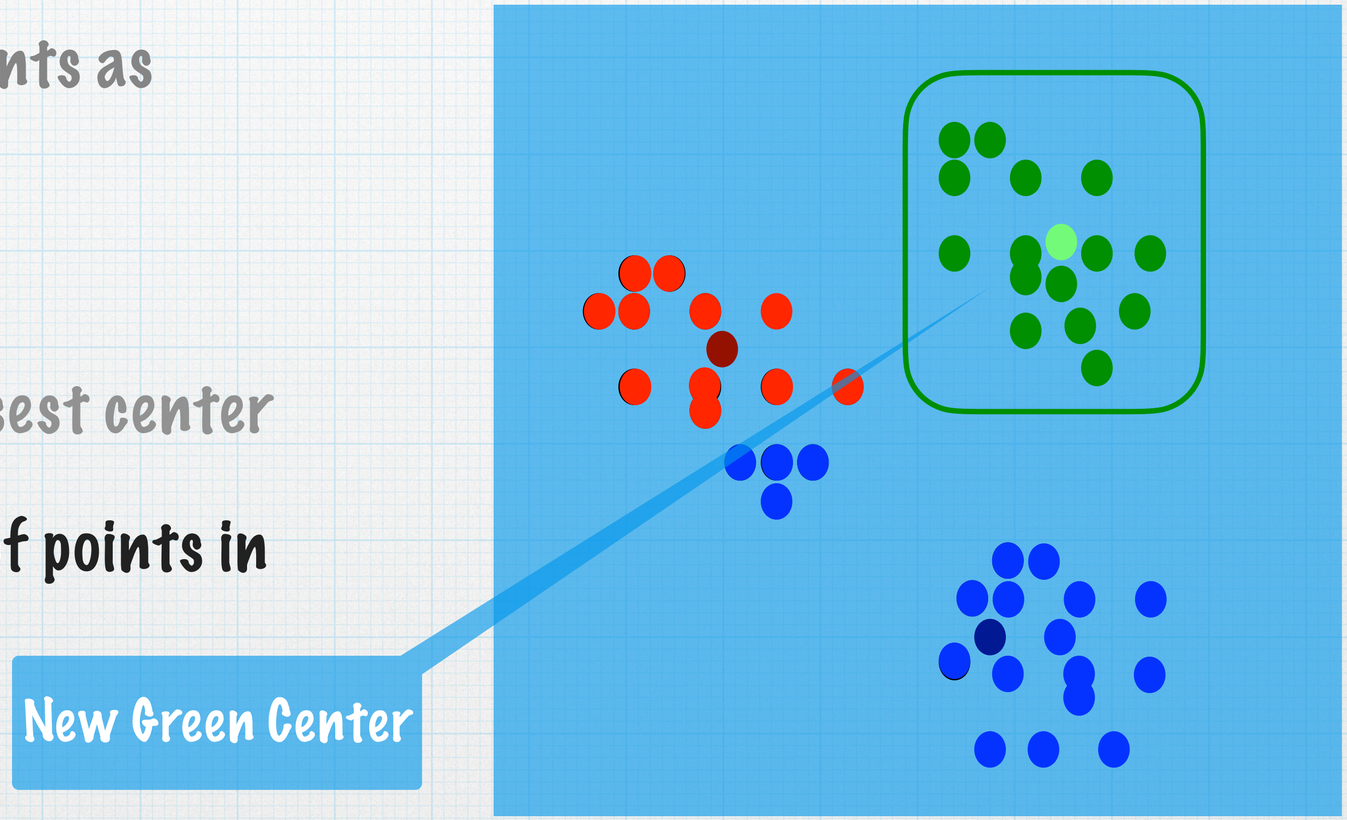
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * Assign data points to closest center
  * **Move center to average of points in cluster**
* Stop when no changes



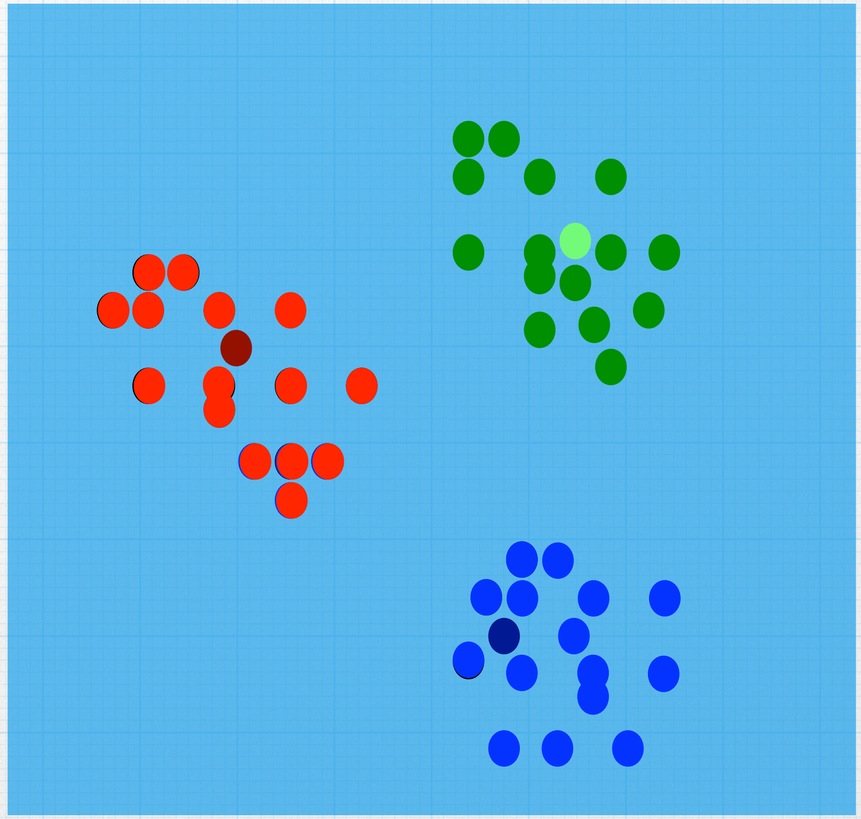
k-Means Algorithm

* Initialize: Pick k random points as cluster centers
* Alternate:
  * Assign data points to closest center
  * **Move center to average of points in cluster**
* Stop when no changes



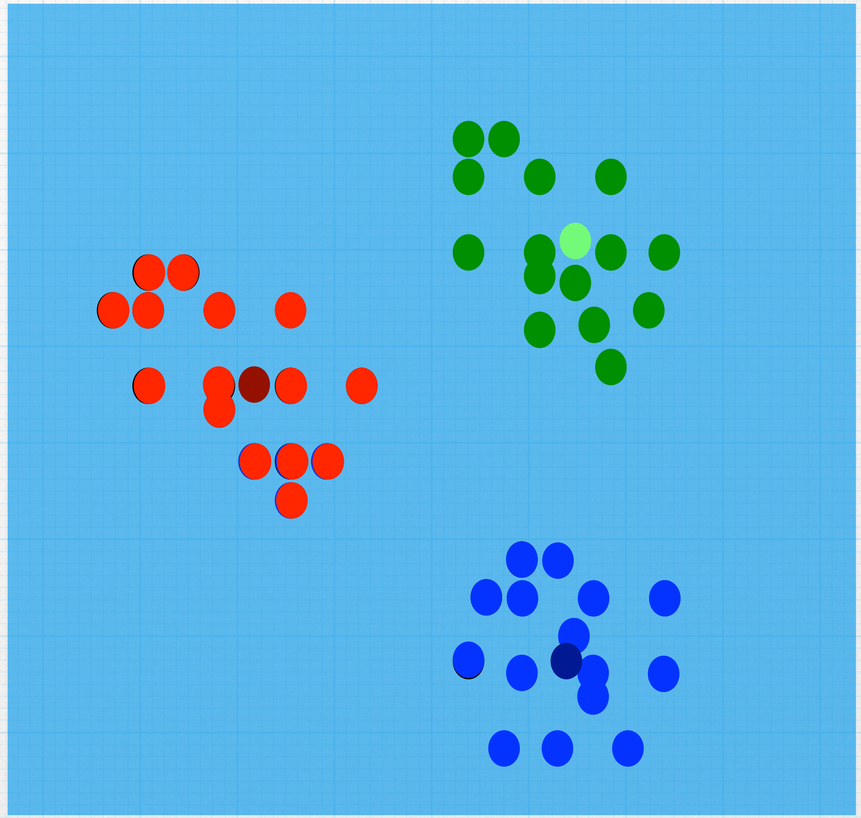
k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - **Assign data points to closest center**
  - Move center to average of points in cluster
- Stop when no changes



k-Means Algorithm

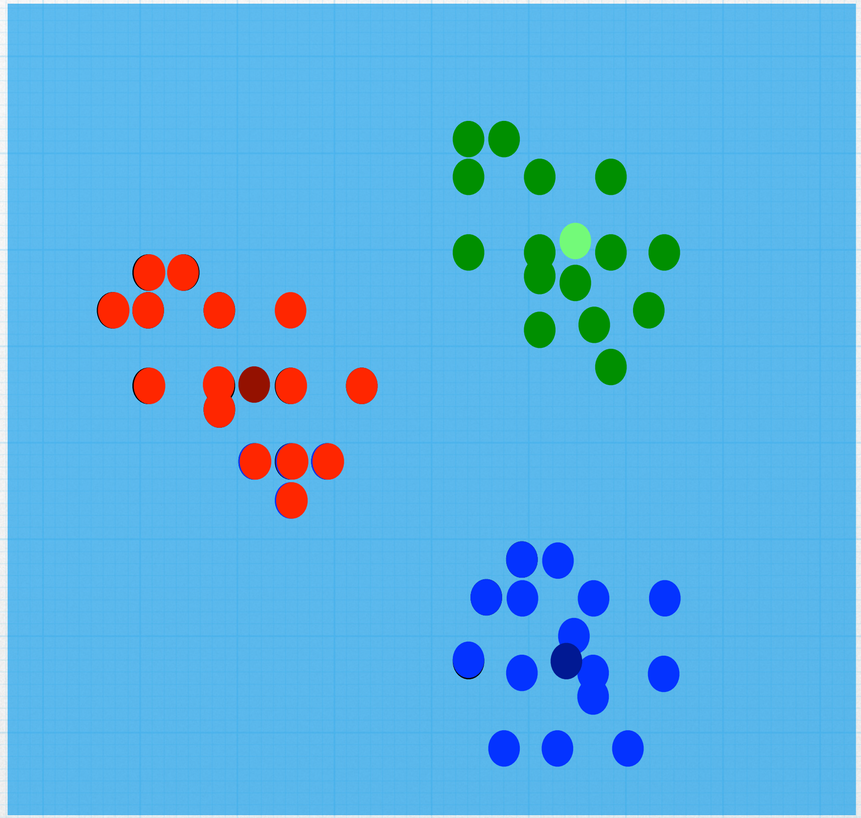
* Initialize: Pick k random points as cluster centers
* Alternate:
  * Assign data points to closest center
  * **Move center to average of points in cluster**
* Stop when no changes



k-Means Algorithm

- Initialize: Pick k random points as cluster centers
- Alternate:
  - Assign data points to closest center
  - Move center to average of points in cluster
- Stop when no changes

Final Clustering!



Code Block A: Implementing k-Means

Identifying Structure in Data

* Data seems very high-dimensional but is it really?
* Can we identify the important directions?

Example: Checks vs Dollars

* Design a smart ATM Camera: Checks vs Dollars
* Dataset: 87 images. Each: $322 \times 137 \times 3$

Example: Checks vs Dollars

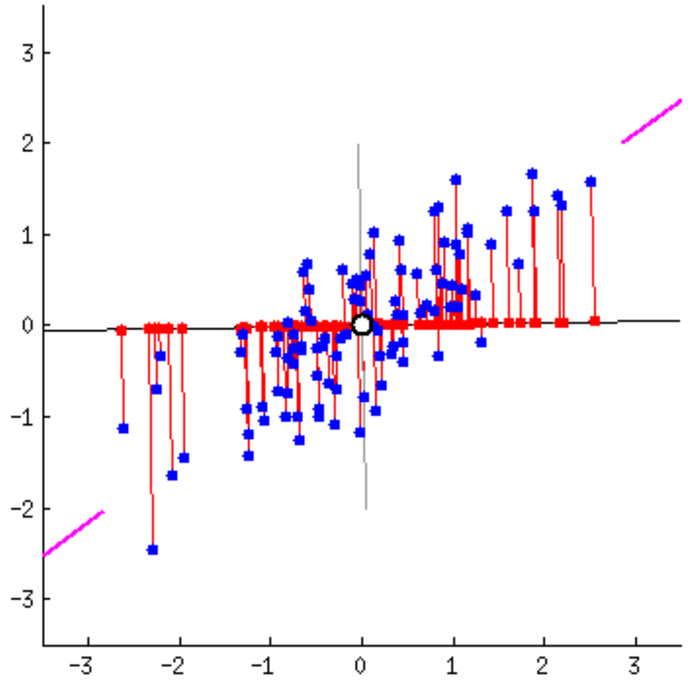
* Design a smart ATM Camera: Checks vs Dollars
* Dataset: 87 images. Each: $322 \times 137 \times 3$
* How many features? 136452!
* Hopeless with only 87 images ...
  * Need to reduce features

Principal Component Analysis

* PCA is an unsupervised technique
* Step 1: Identify the direction in which data varies the most. This is the first principal component
* Step 2: Look for a perpendicular direction along which data varies second most: Second principal component
* Repeat for third principal component, ...

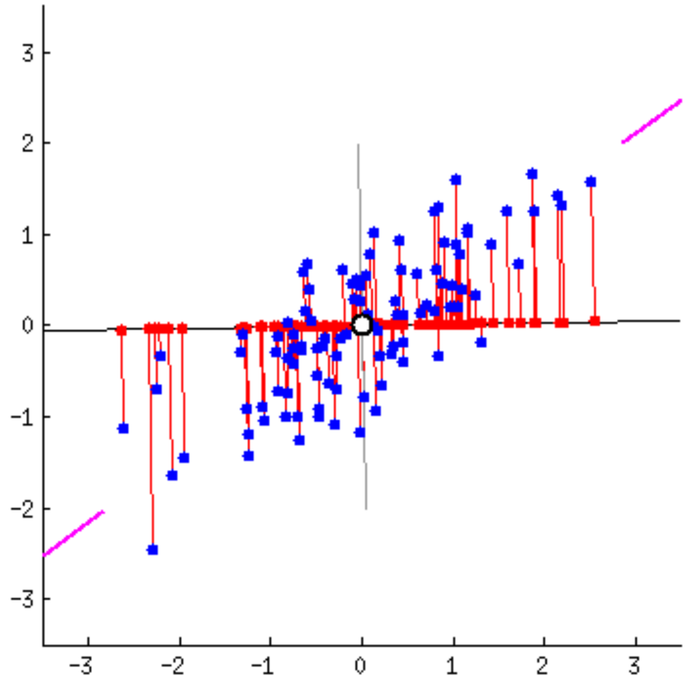
Principal Component Analysis

* Idea: Principal components identify directions along with data vary the most in descending order



Principal Component Analysis

* Idea: Principal components identify directions along with data vary the most in descending order

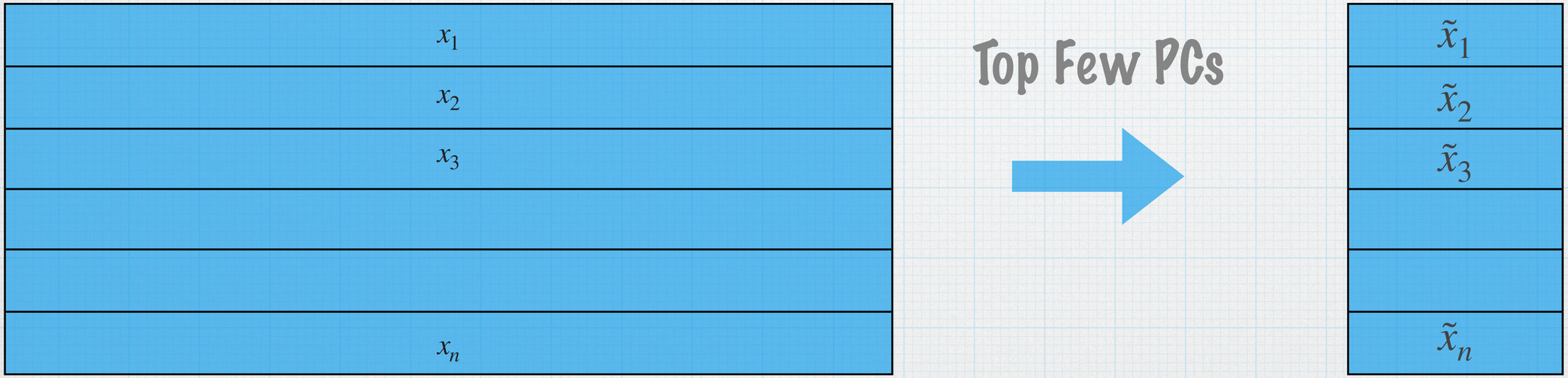


Principal Component Analysis

* Idea: Principal components identify directions along with data vary the most in descending order
* Feature selection/Dimension reduction: Keep only the top principal components
* Once this is done, you can proceed with any data analysis as before

Principal Component Analysis

* Idea: Principal components identify directions along with data vary the most in descending order



High-Dimensional Dataset

Low-Dimensional Dataset

Principal Component Analysis

* Idea: Principal components identify directions along with data vary the most in descending order

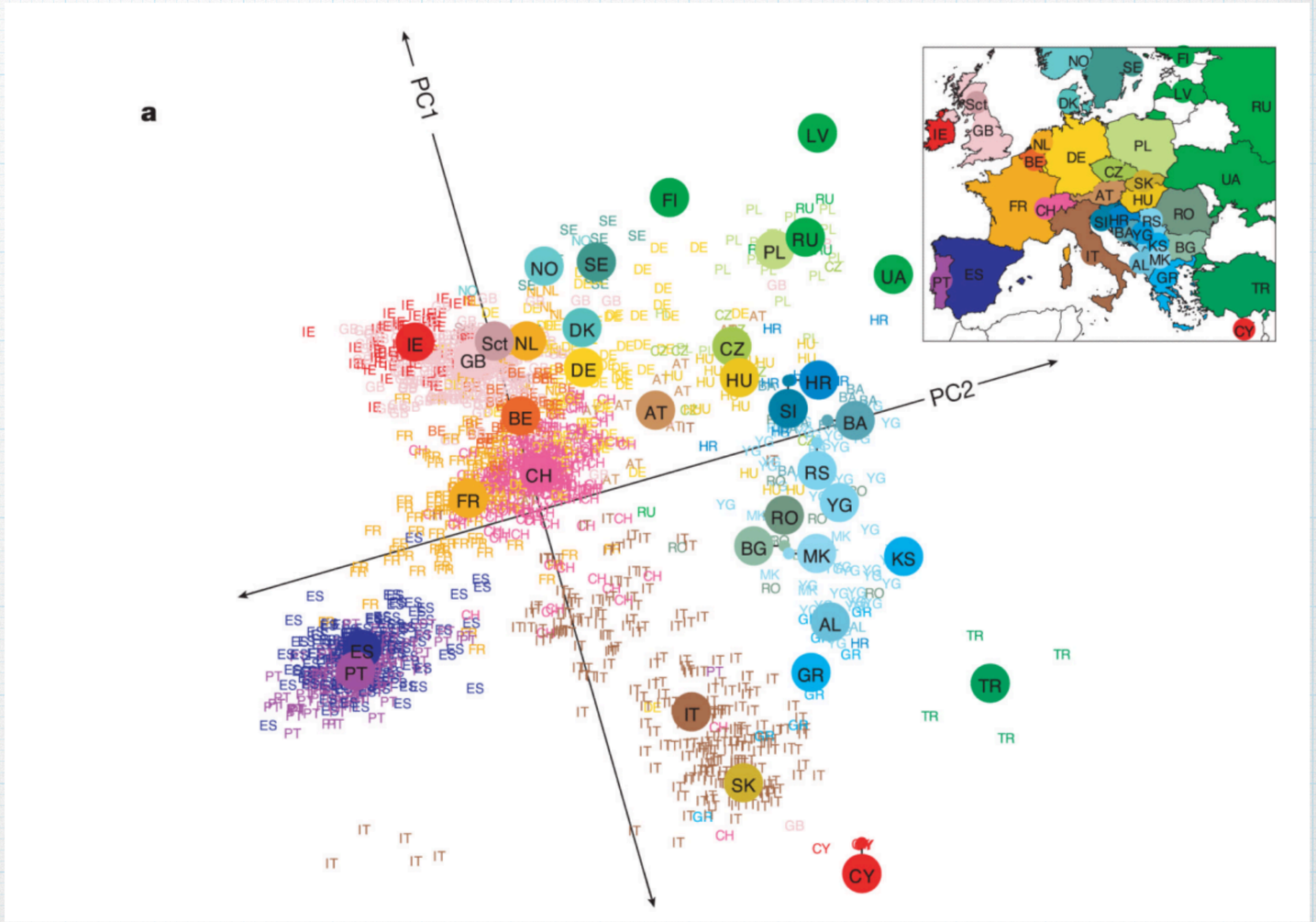
* Applications:
  * Compression of data
  * Better accuracy in downstream tasks
  * Reveals hidden structure in data

PCA: Genes and Geography

* 23andMe Experiment:
  * Collect genetic sequencing data of 6000 individuals
  * Map gene sequences to vectors - Very high-dimensional (> 100k)
  * Extract top two principal components
  * Plot the 2d-points and color them by country

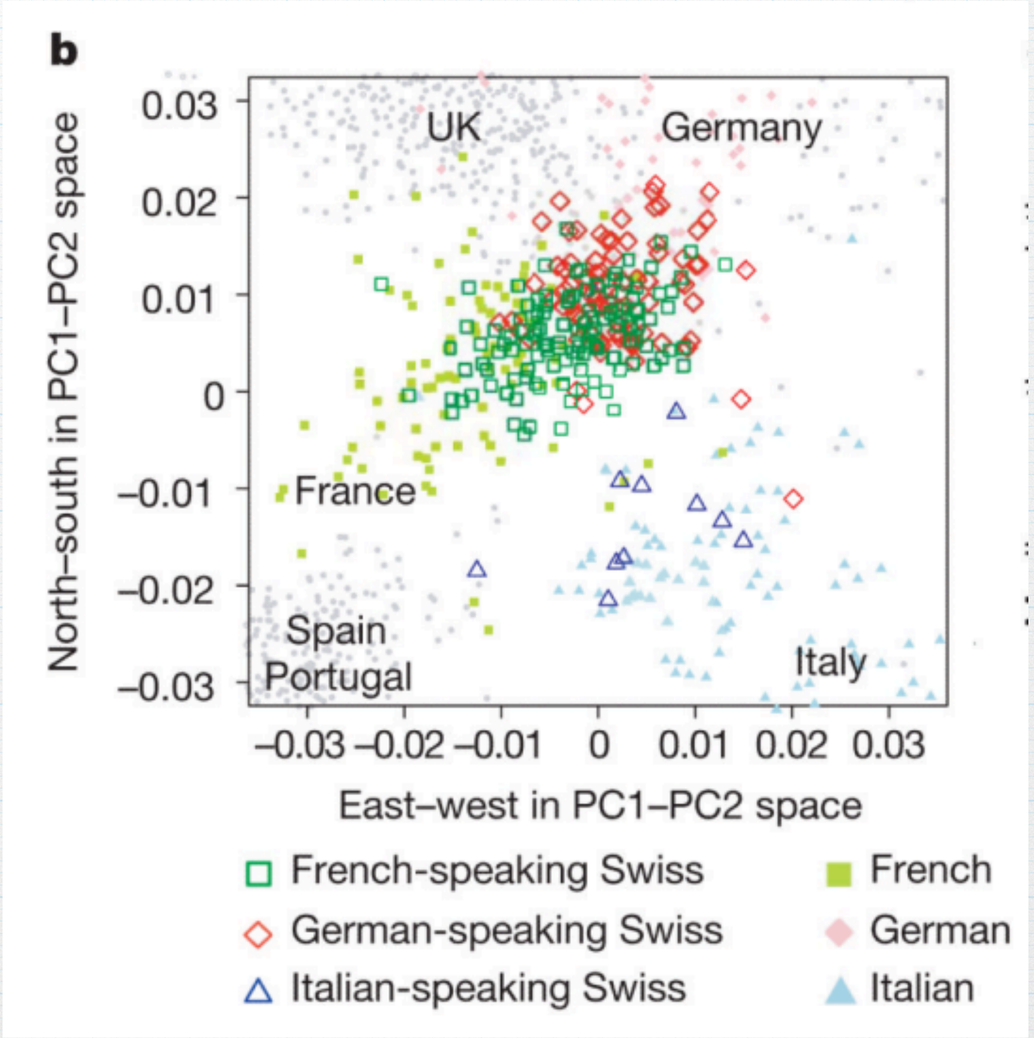
PCA: Genes and Geography

- 23andMe Experiment:
  - Collect genetic sequencing data of 6000 individuals
  - Map gene sequences to vectors - Very high-dimensional (> 100k)
  - Extract top two principal components
  - Plot the 2d-points and color them by country



PCA: Genes and Geography

- 23andMe Experiment:
  - Collect genetic sequencing data of 6000 individuals
  - Map gene sequences to vectors - Very high-dimensional (> 100k)
  - Extract top two principal components
  - Plot the 2d-points and color them by country



Principal Component Analysis

- PCA should often be the first step in doing even supervised learning. Ubiquitous tool
- Applications:
  - Compression of data
  - Better accuracy in downstream tasks
  - Reveals hidden structure in data

Code Block C: Extracting PCs using Scikit

Application of PCA: Document Analysis

* Documents are vectors: Each document as a vector of how many times each word appears in it

Document1 

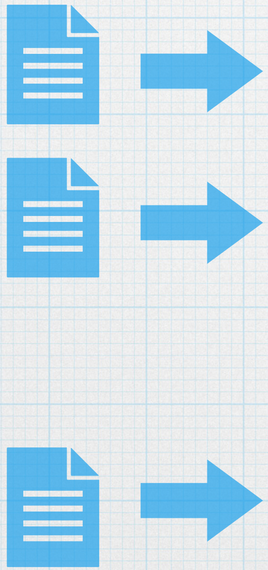
Document2 

...

Documentn 

Application of PCA: Document Analysis

* Documents are vectors: Each document as a vector of how many times each word appears in it

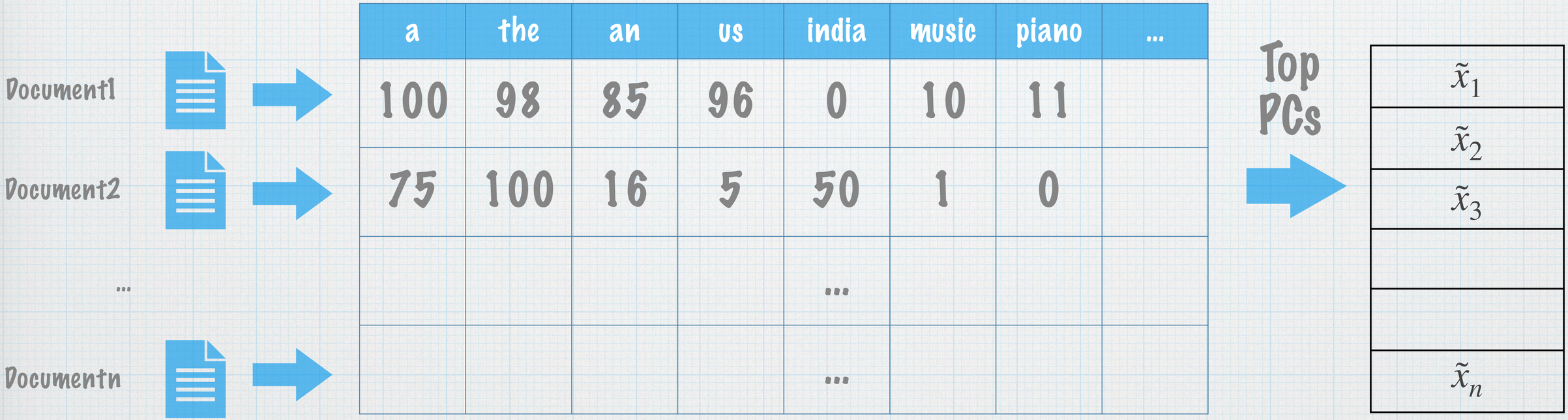


	a	the	an	us	india	music	piano	...
Document1	100	98	85	96	0	10	11	
Document2	75	100	16	5	50	1	0	
...					...			
Documentn					...			

Each document is a high-dimensional vector!

Application of PCA: Document Analysis

* Documents are vectors: Each document as a vector of how many times each word appears in it



Each document is a high-dimensional vector!

Do PCA!

Code Block D: Document Analysis using PCA

Summary

* Even without labels we can learn a lot from data
* Clustering: Group similar objects together
* PCA: Feature extraction/dimension-reduction/ interpreation

Future ... Representation Learning

* Extract powerful representations for downstream tasks
* Revolutionized imaging, translation, images->text, Natural Language Processing, every modern ML ...

THANK YOU