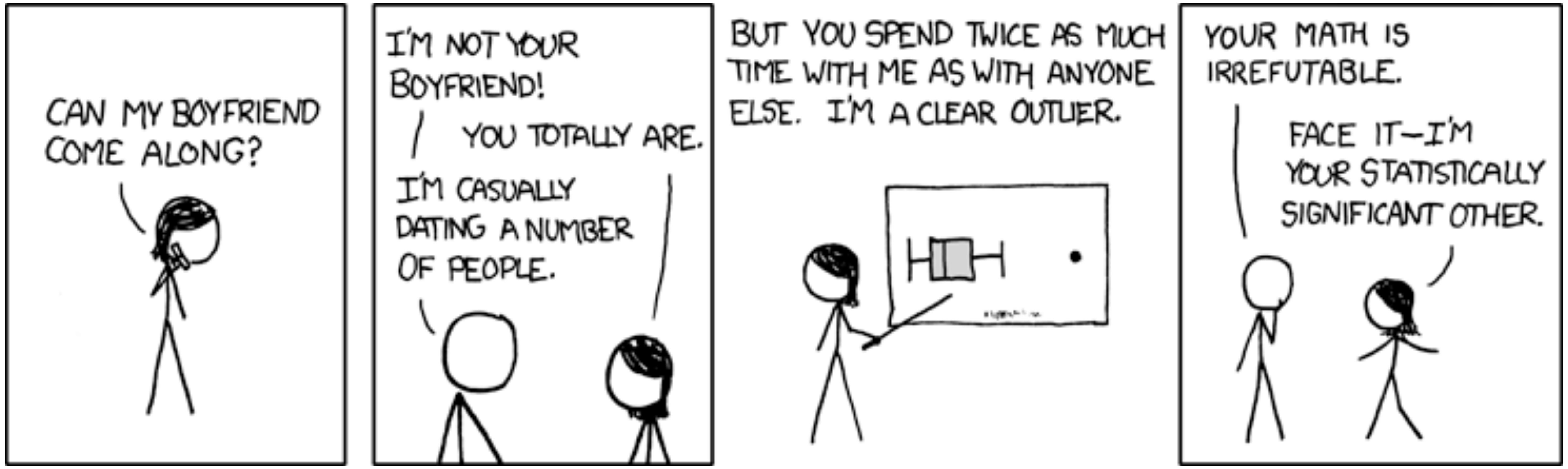
CS-5630 / CS-6630 Visualization for Data Science
Filtering & Aggregation

Alexander Lex
alex@sci.utah.edu

THE UNIVERSITY OF UTAH



[xkcd]

Reducing Items and Attributes

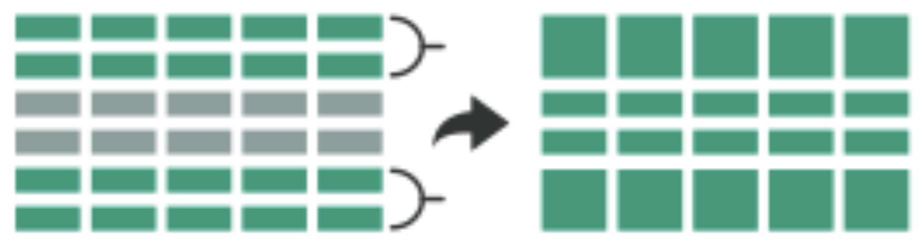
Filter

→ Items

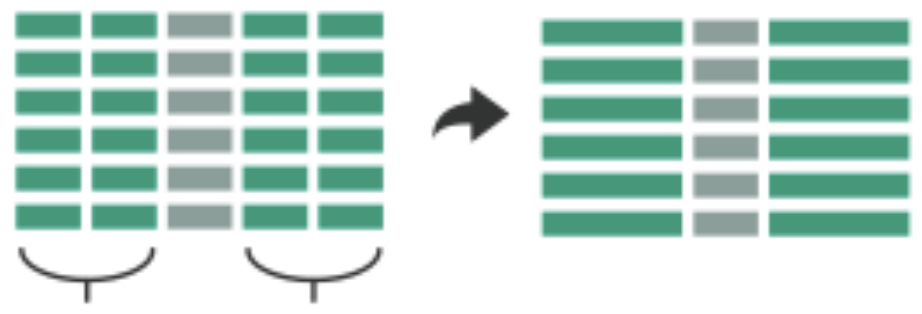
→ Attributes

Aggregate

→ Items



→ Attributes



Filter

elements are eliminated

What drives filters?

Any possible function that partitions a dataset into two sets

- Bigger/smaller than x
- Fold-change
- Noisy/insignificant

Dynamic Queries / Filters

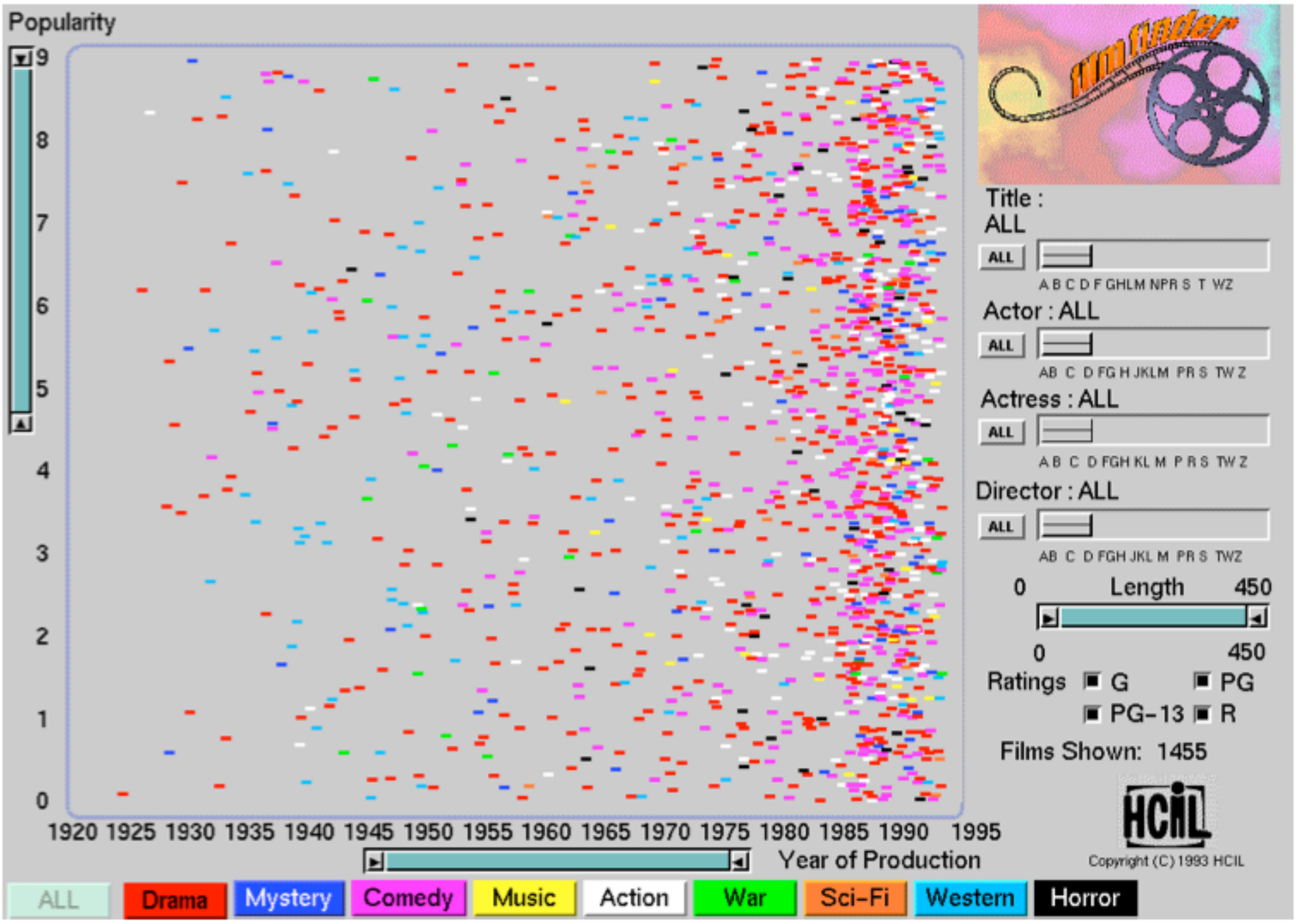
coupling between encoding and interaction so that user can immediately see the results of an action

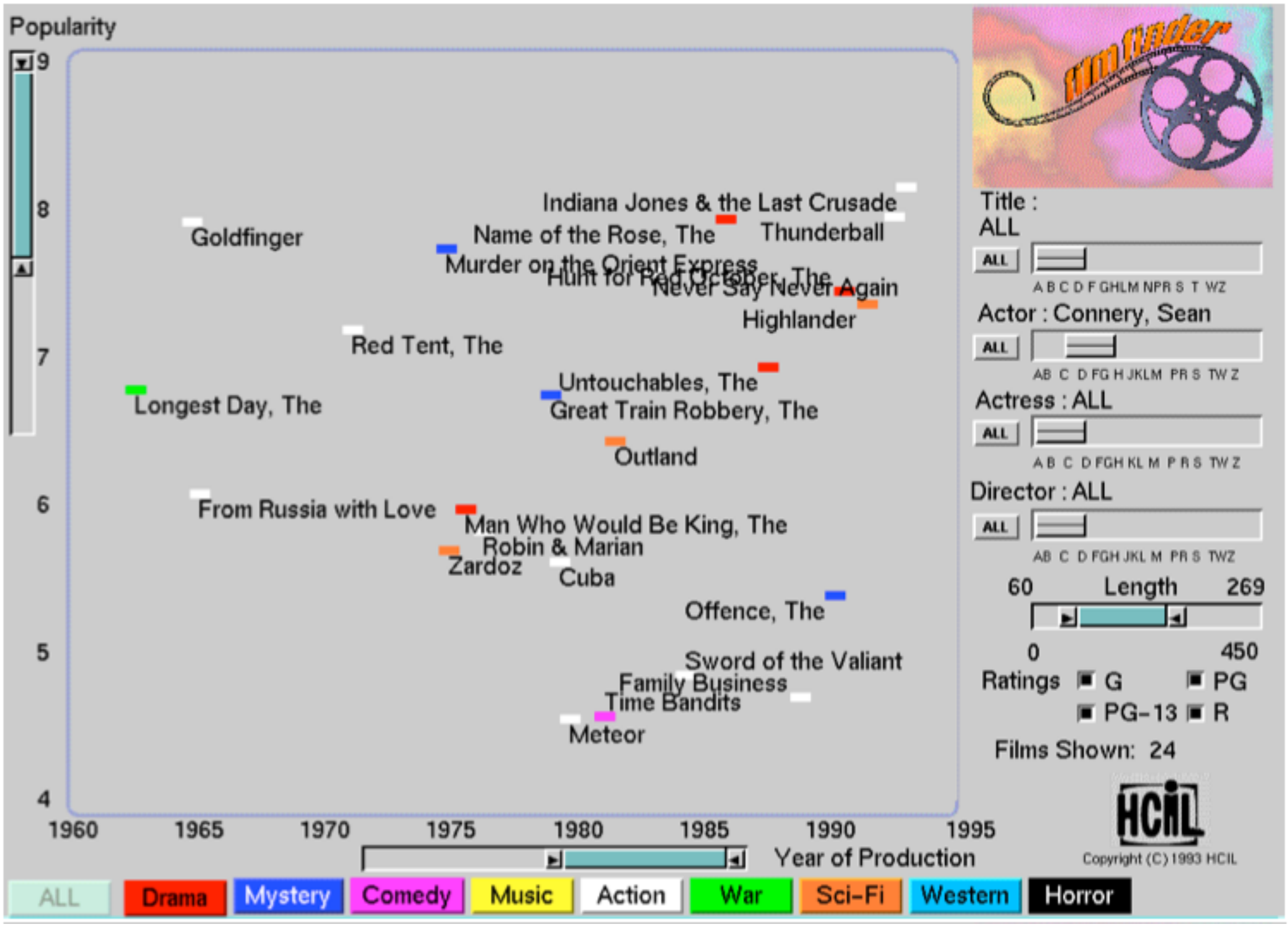
Queries: start with 0, add in elements

Filters: start with all, remove elements

Approach depends on dataset size

ITEM FILTERING



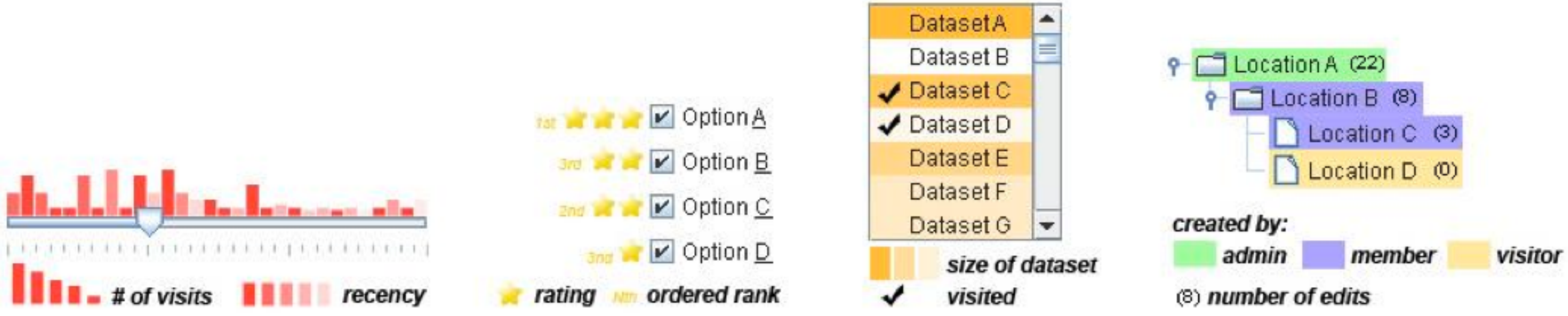


Ahlberg 1994

Scented Widgets

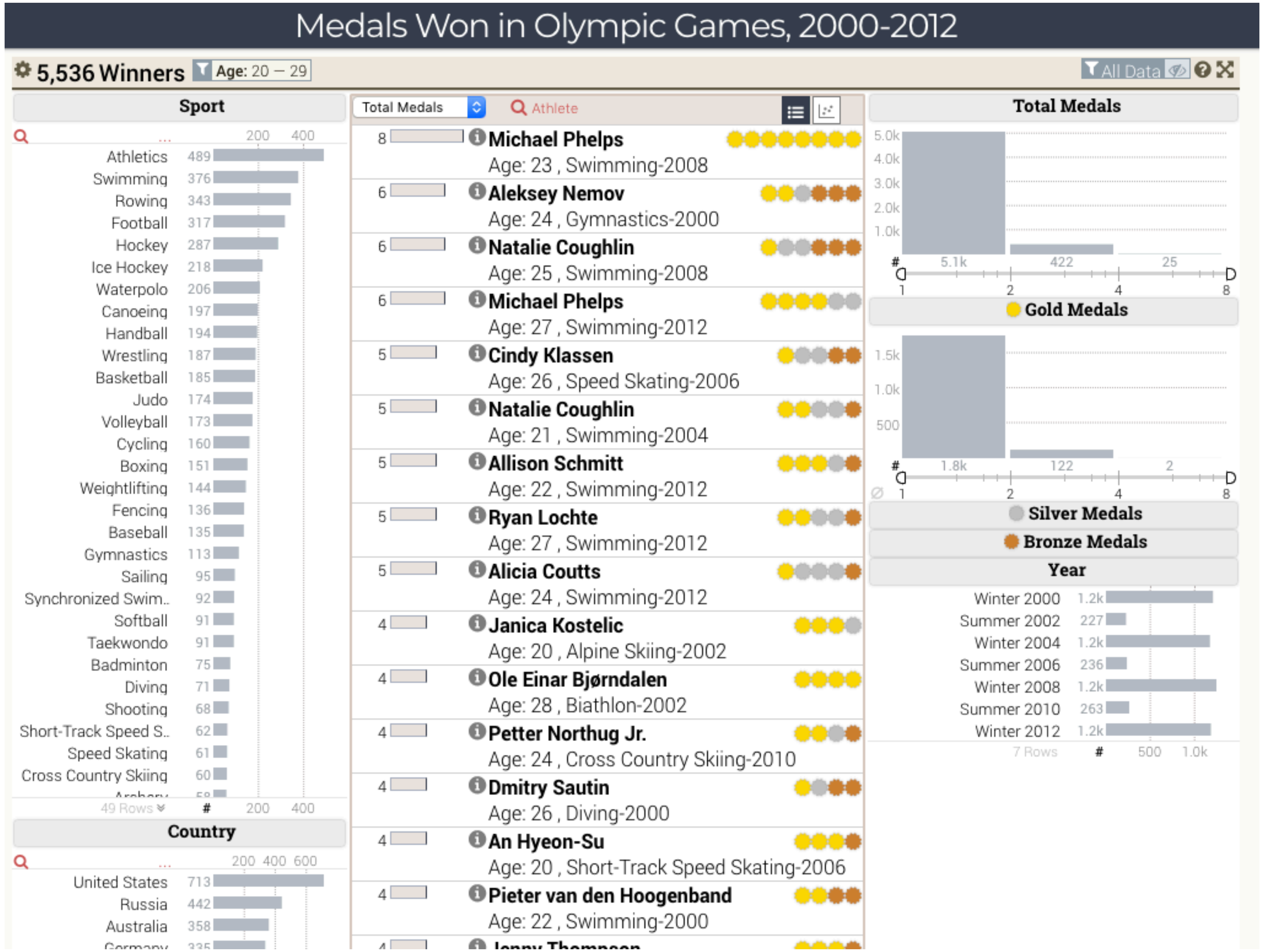
information scent: user's (imperfect) perception of data

GOAL: lower the cost of information foraging through better cues



Willett 2007

Item Filtering with Scented Widgets

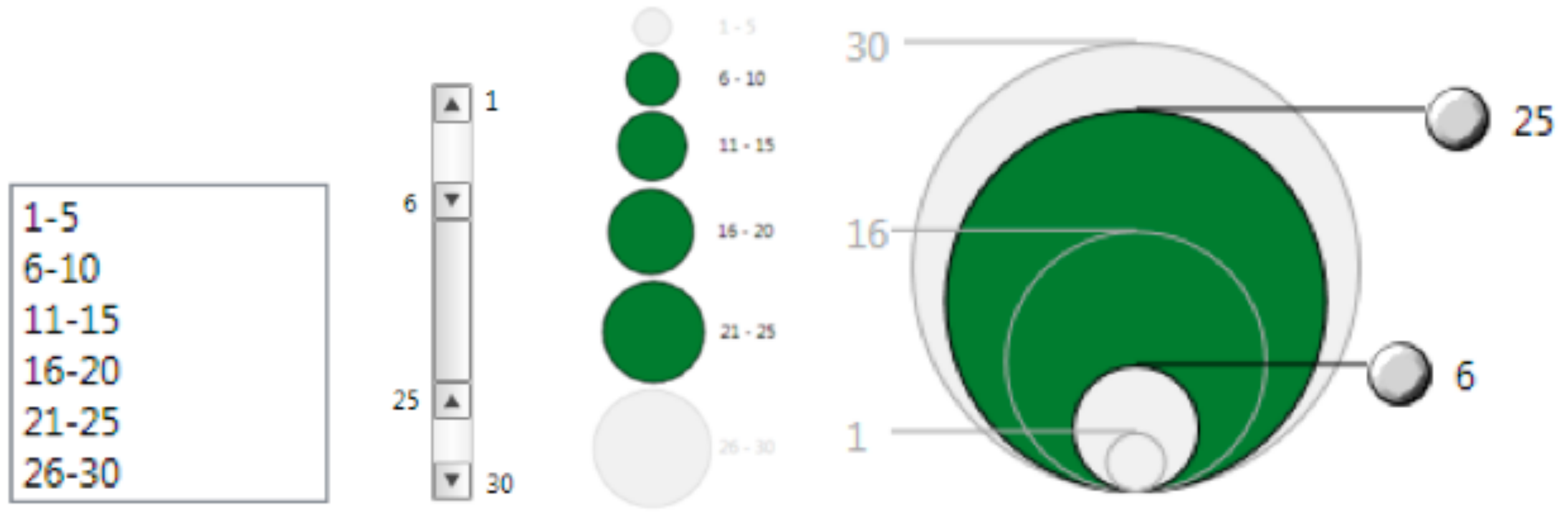


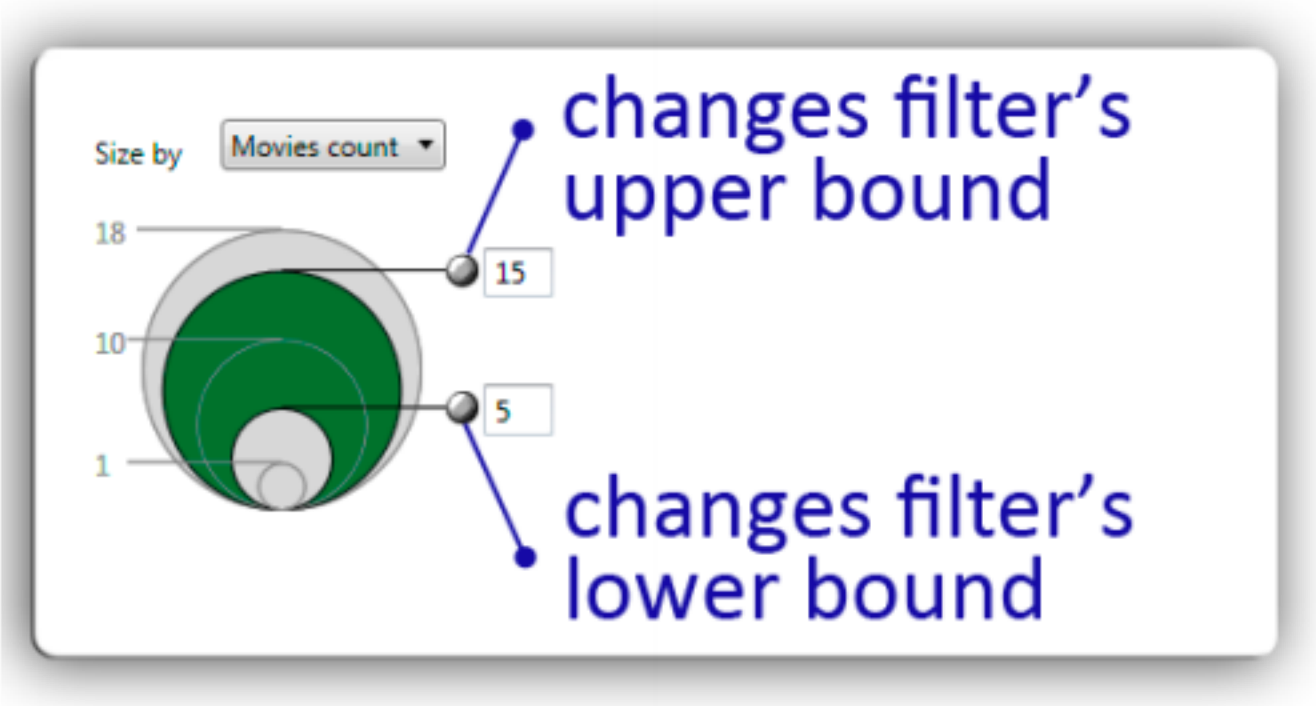
https://keshif.me/gallery/olympics

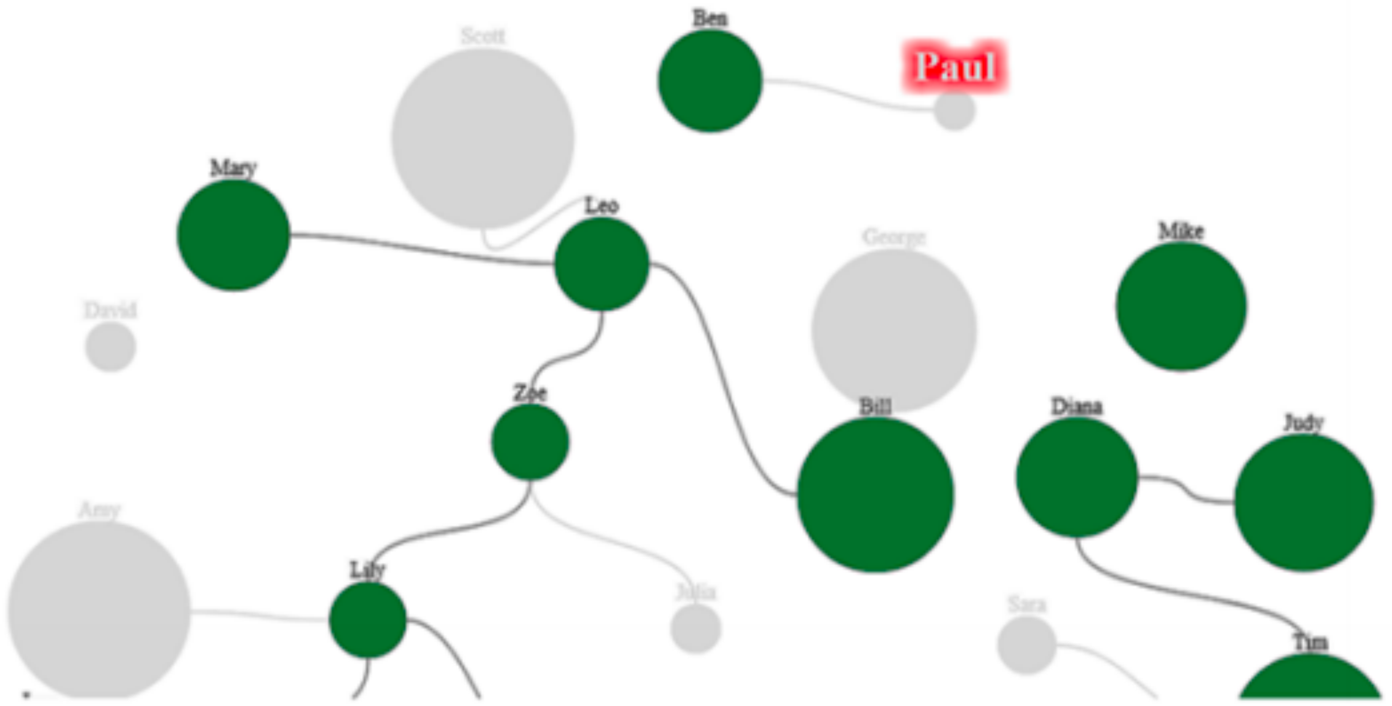
Interactive Legends

Controls combining the visual representation of static legends with interaction mechanisms of widgets

Define and control visual display together

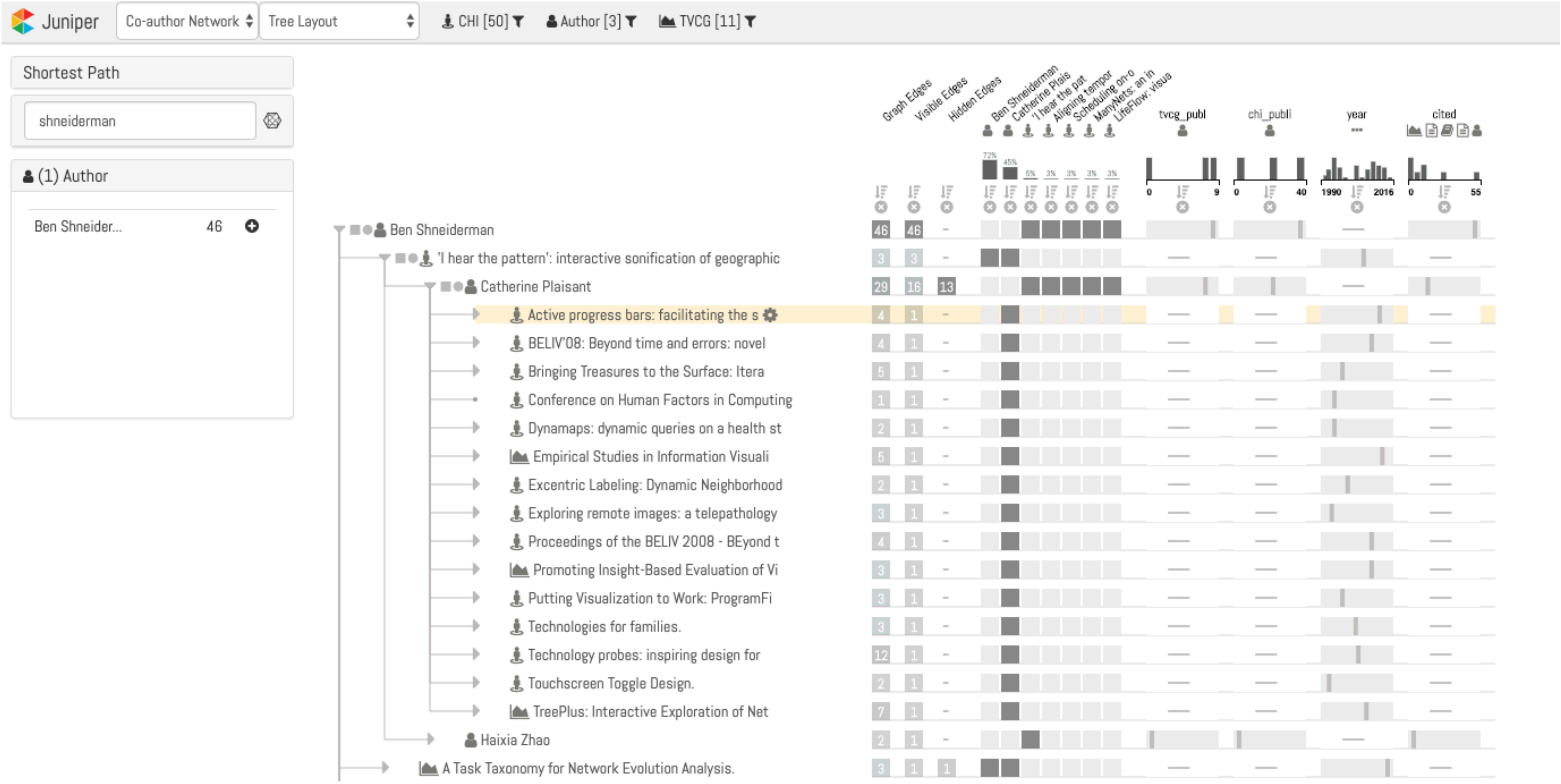






Riche 2010

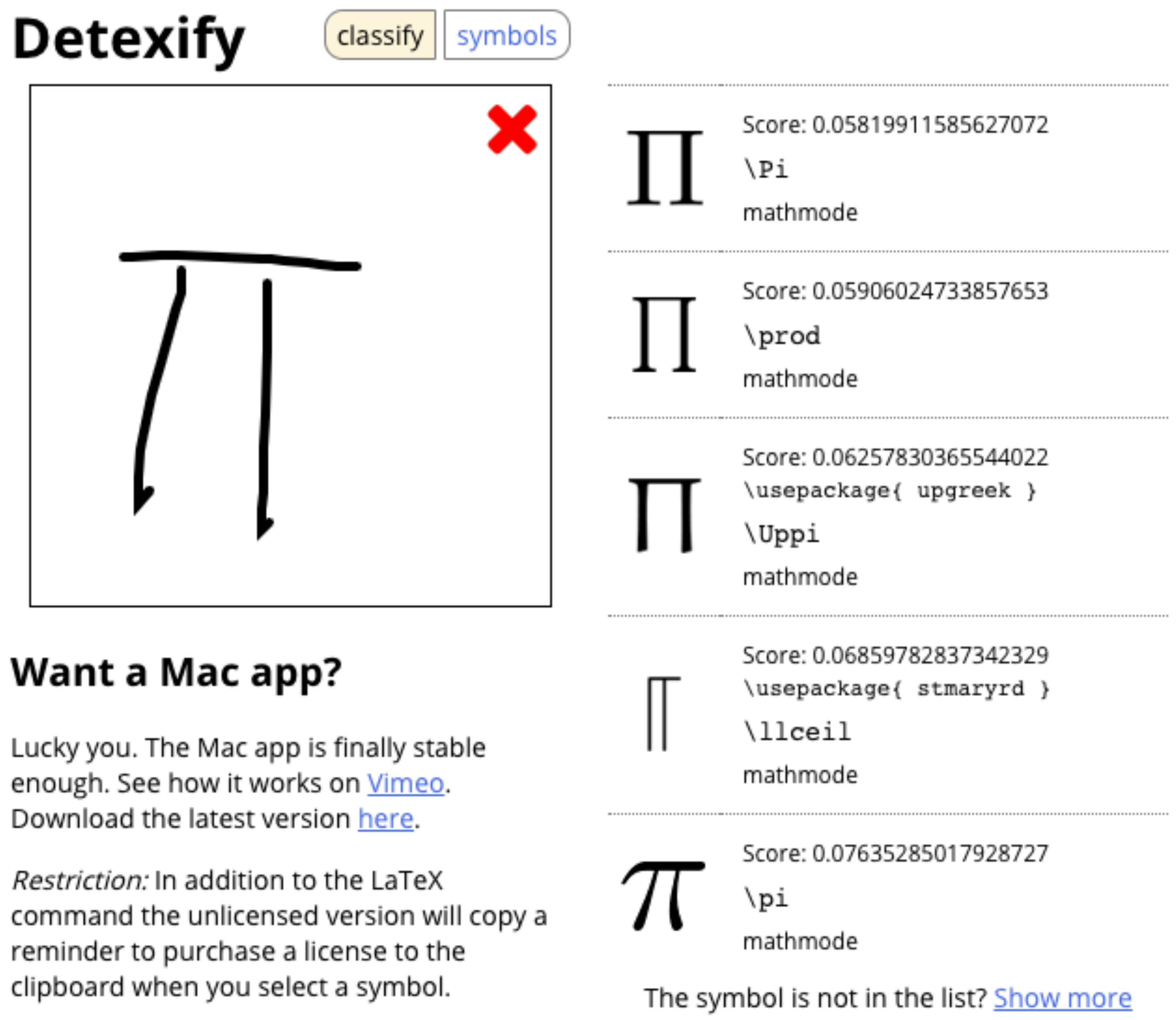
Text & Dynamic Queries



Sketch-based Queries

Idea: we have a mental model of a pattern.

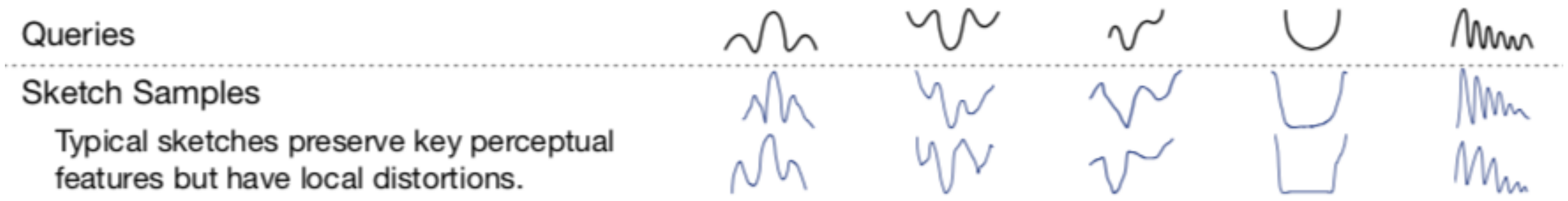
Let user sketch it!



http://detexify.kirelabs.org/classify.html

Sketch-based Queries

Time Series



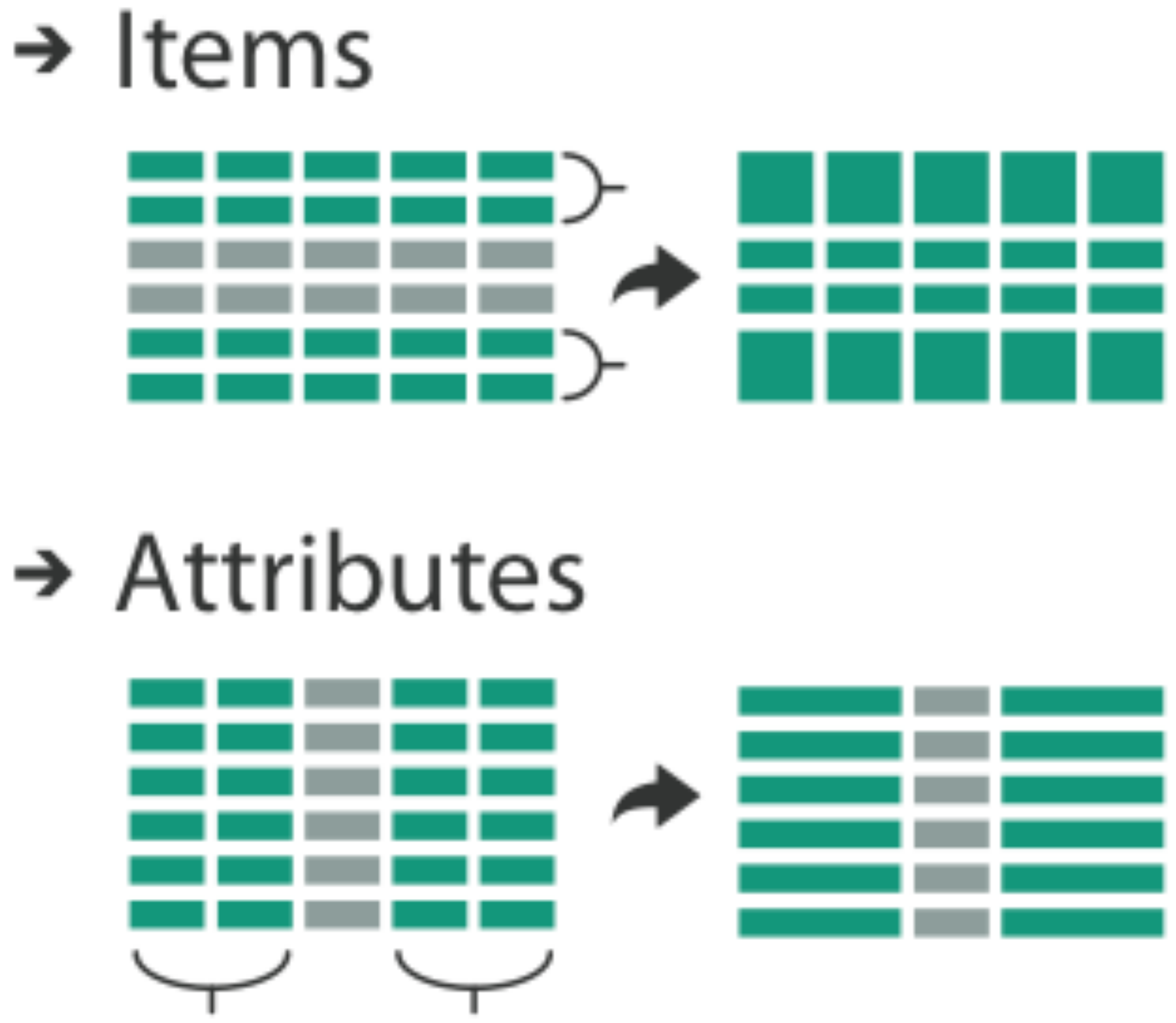
https://www.youtube.com/watch?v=4YQTuUuIFbI

[Mannino, Abouzied, 2018]

Aggregation

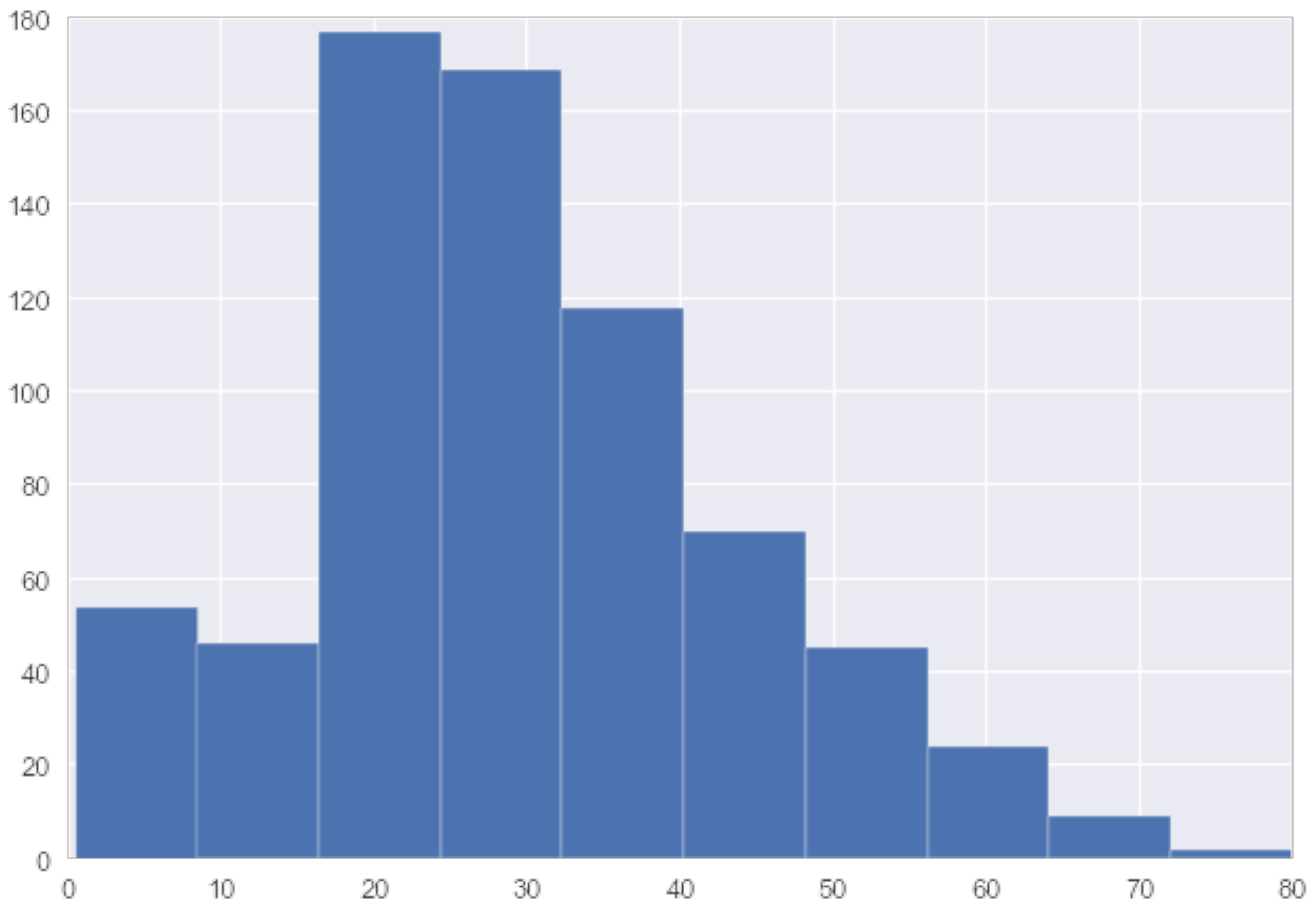
Aggregate

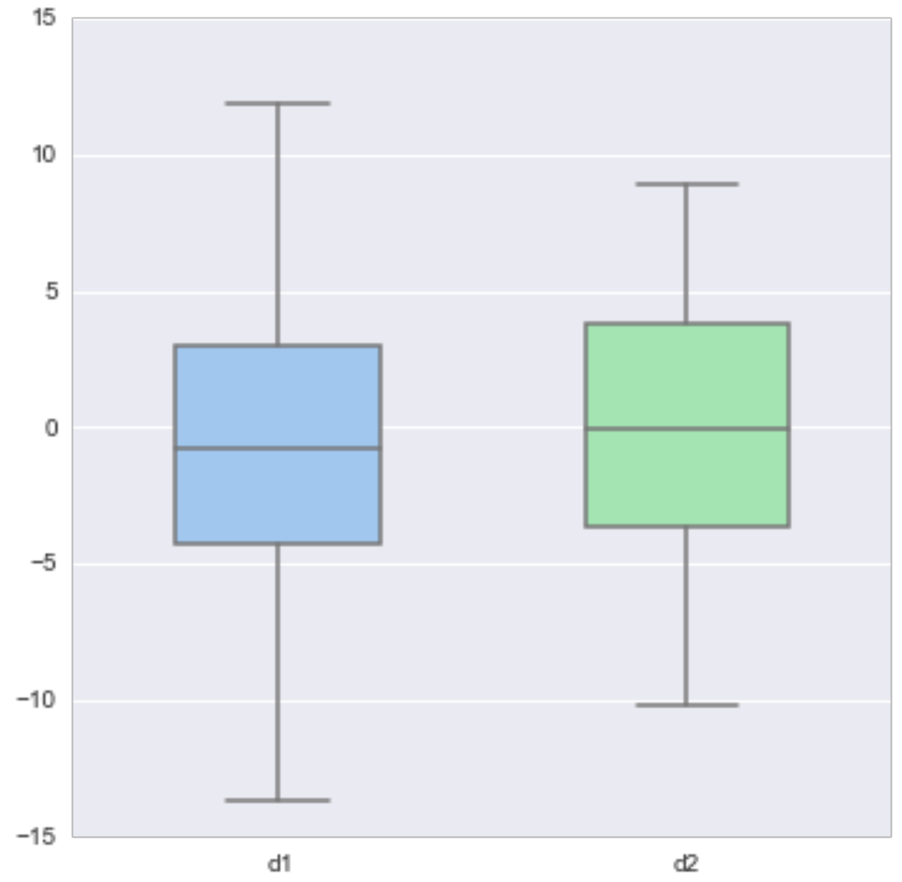
a group of elements is represented by a (typically smaller) number of derived elements

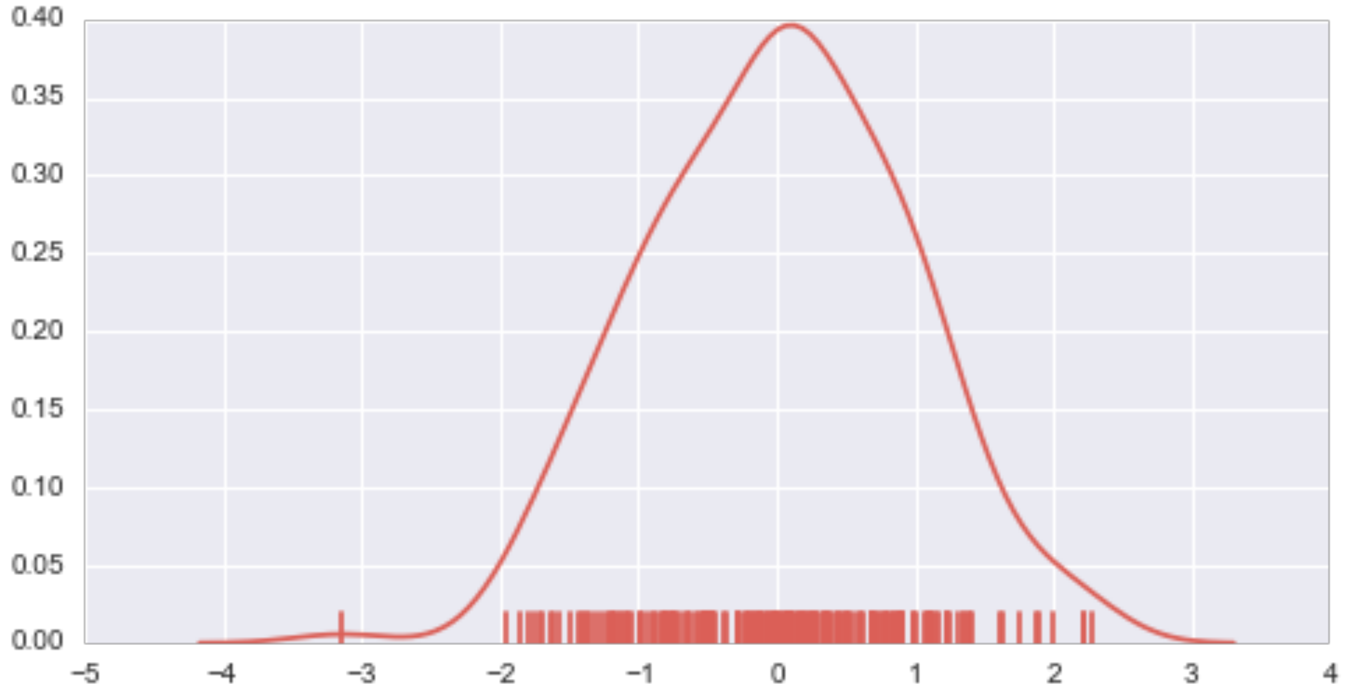


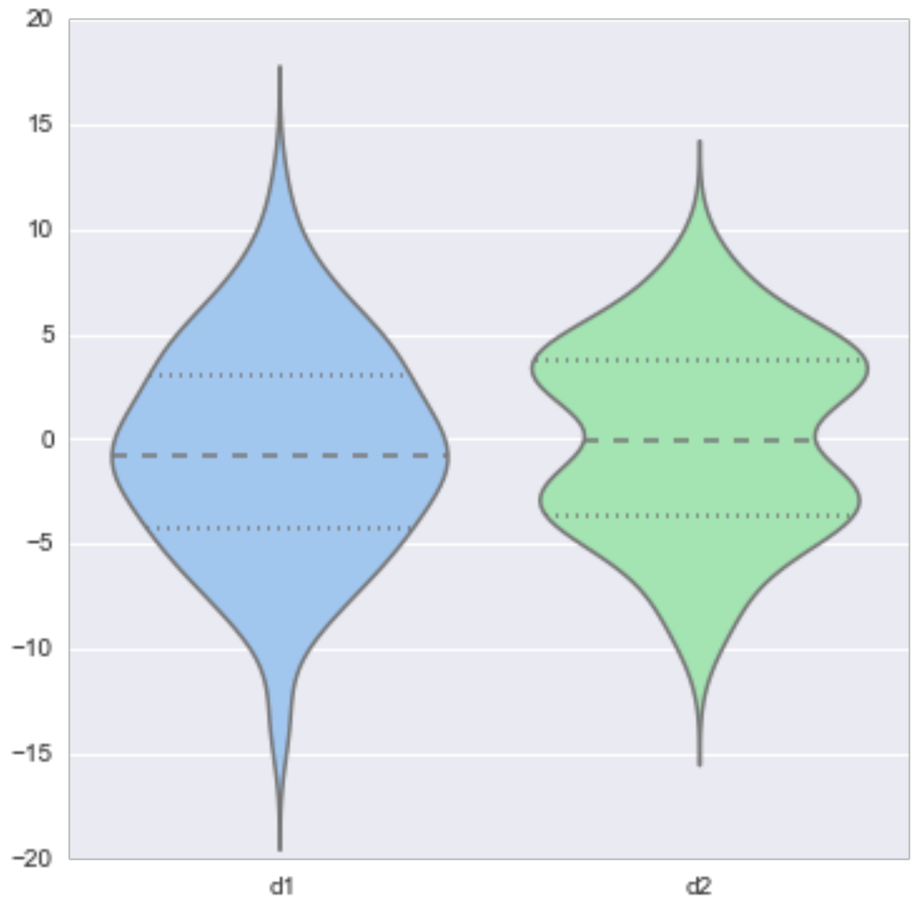
Why Aggregate?

Recall Tabular Aggregation





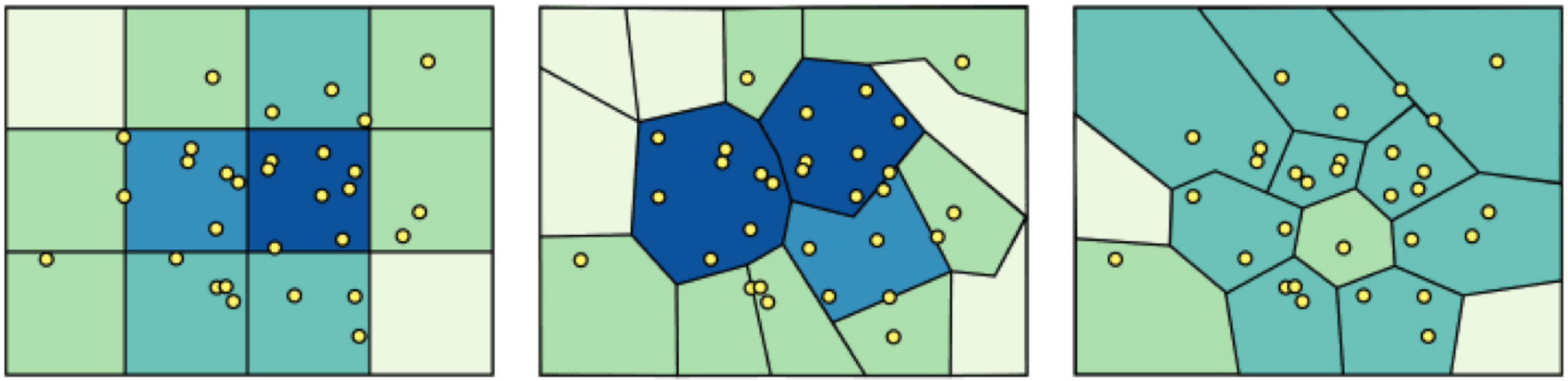


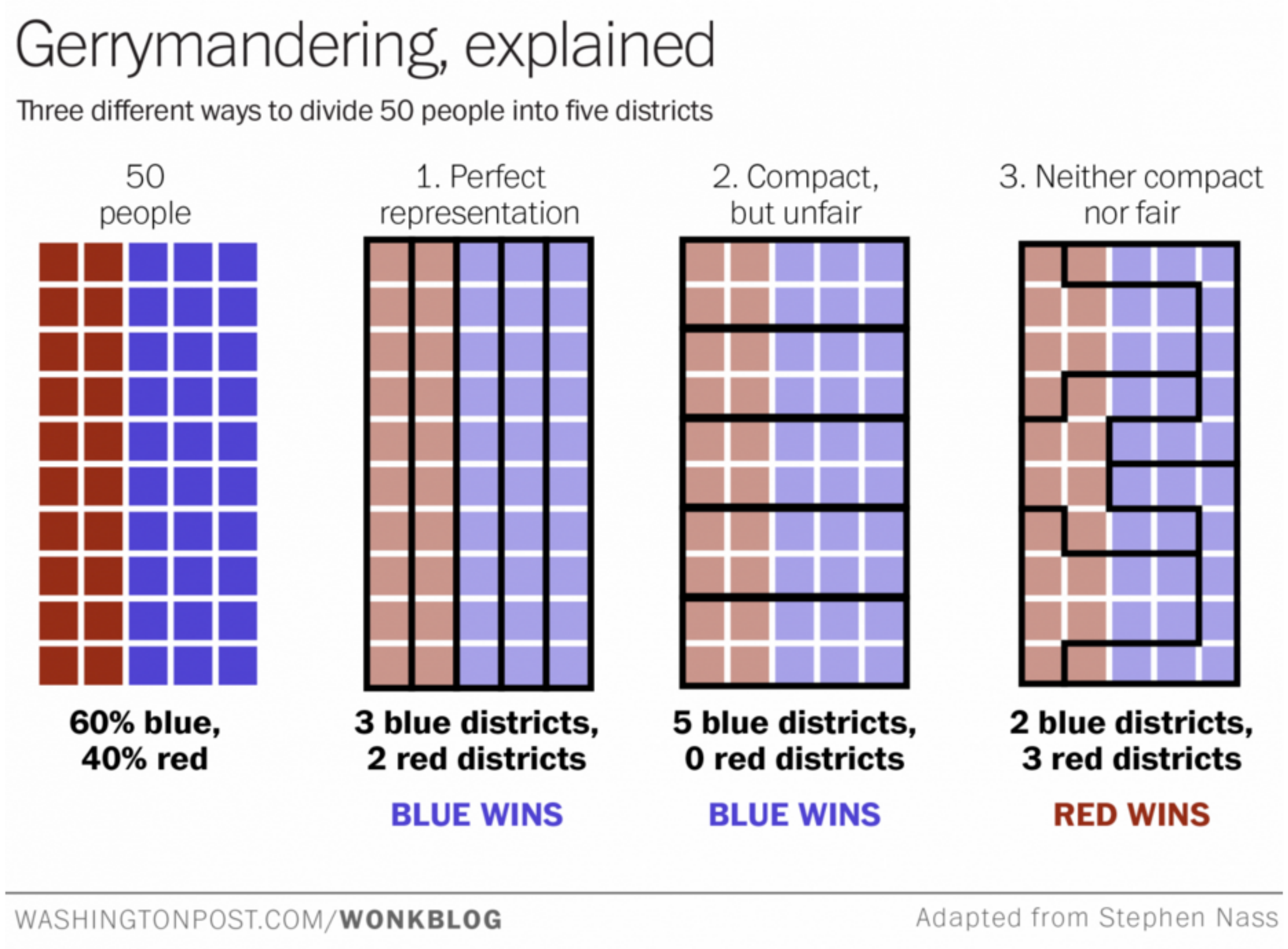


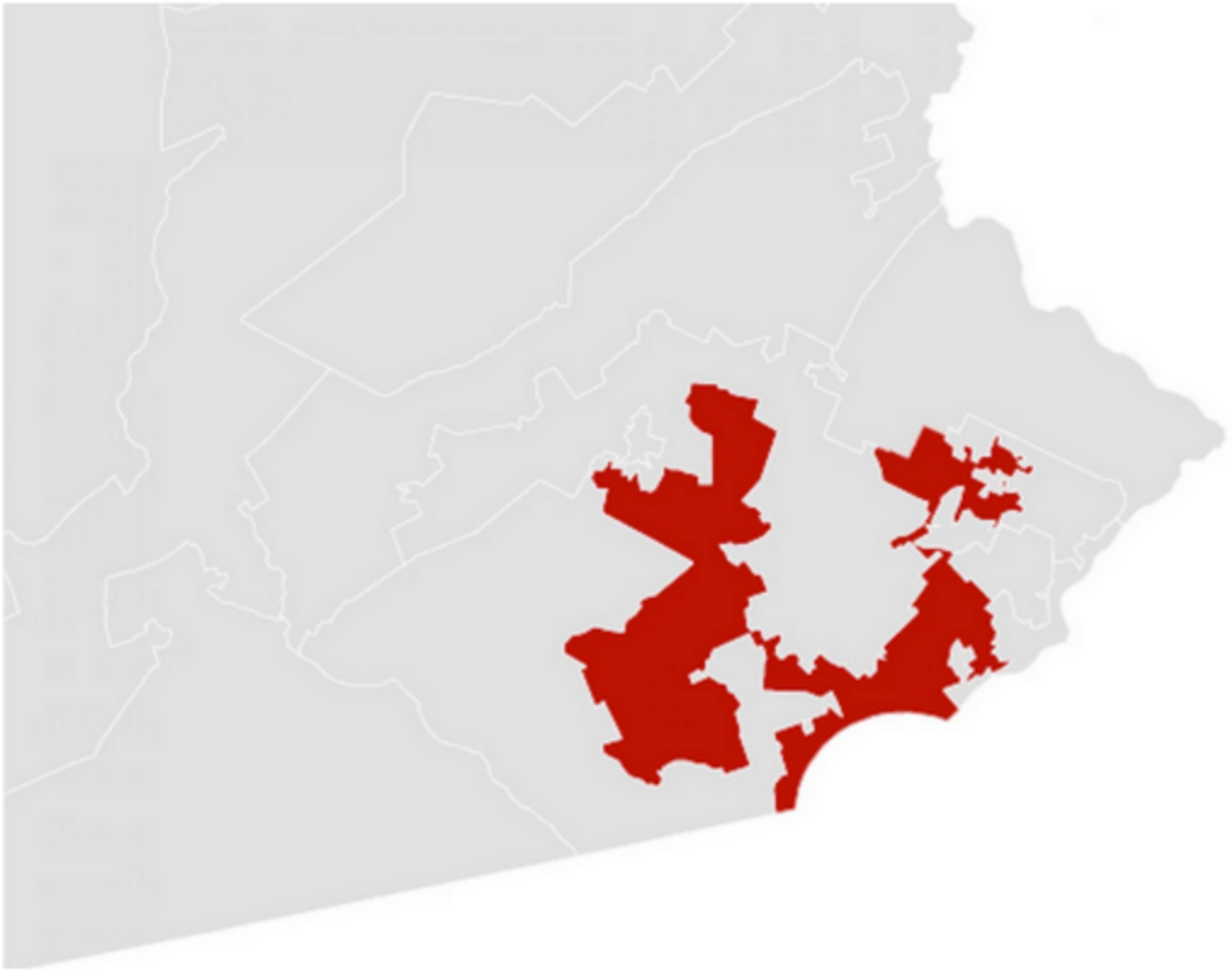
Spatial Aggregation

modifiable areal unit problem

in cartography, changing the boundaries of the regions used to analyze data
can yield dramatically different results

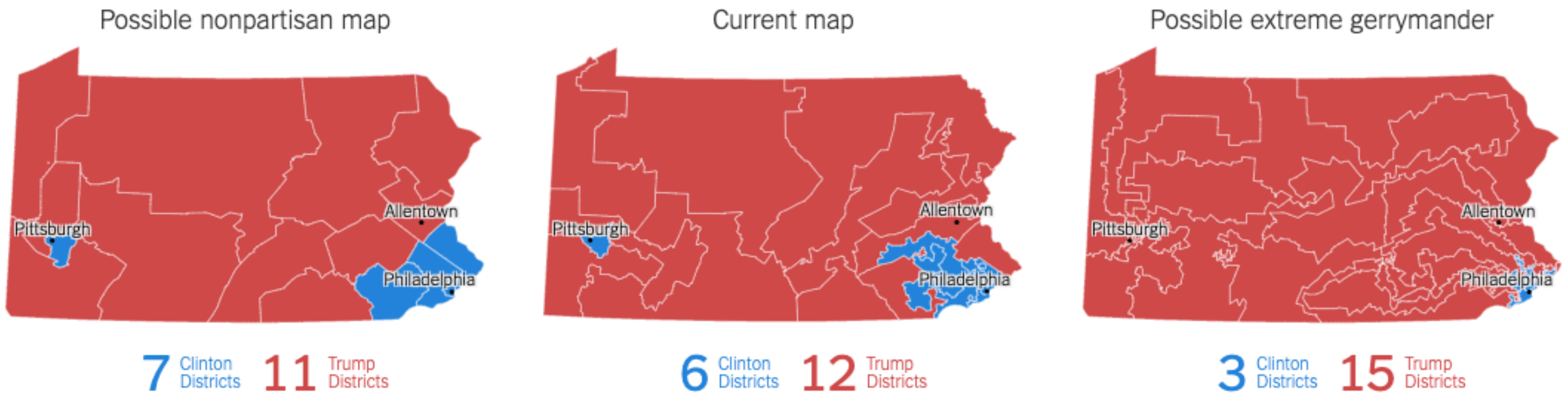




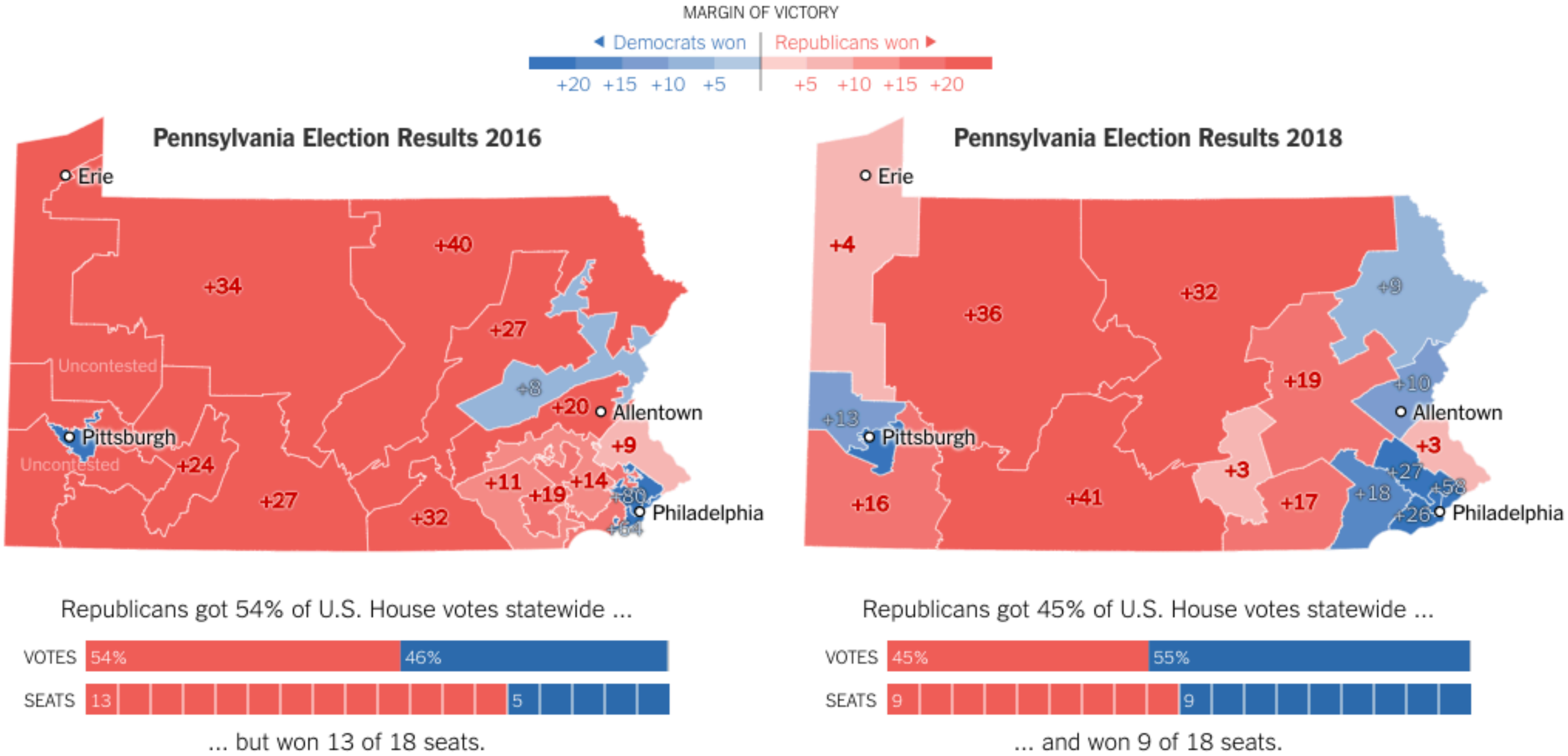


A real district in Pennsylvania
Democrats won 51% of the vote
but only 5 out of 18 house seats

Gerrymandering in PA

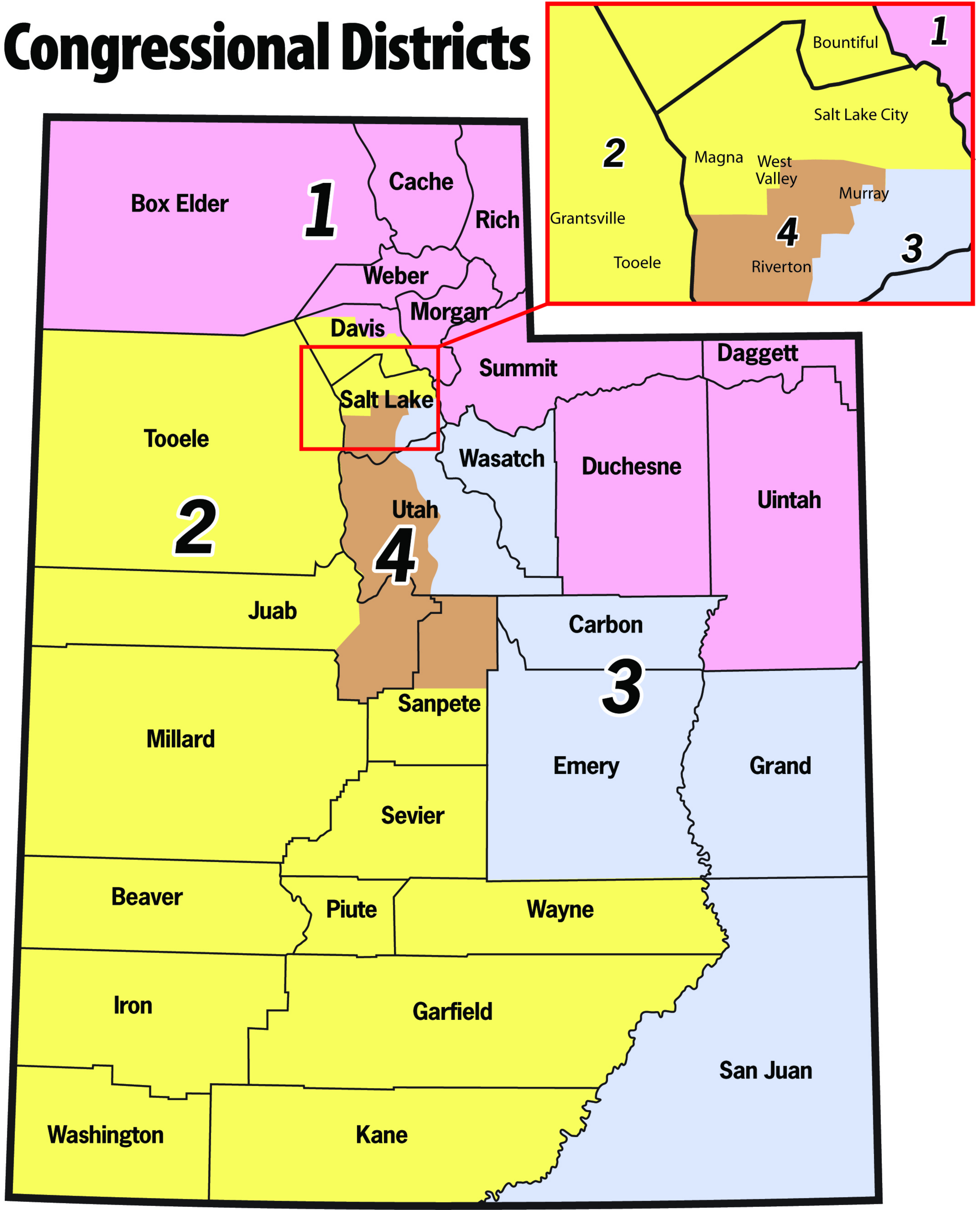


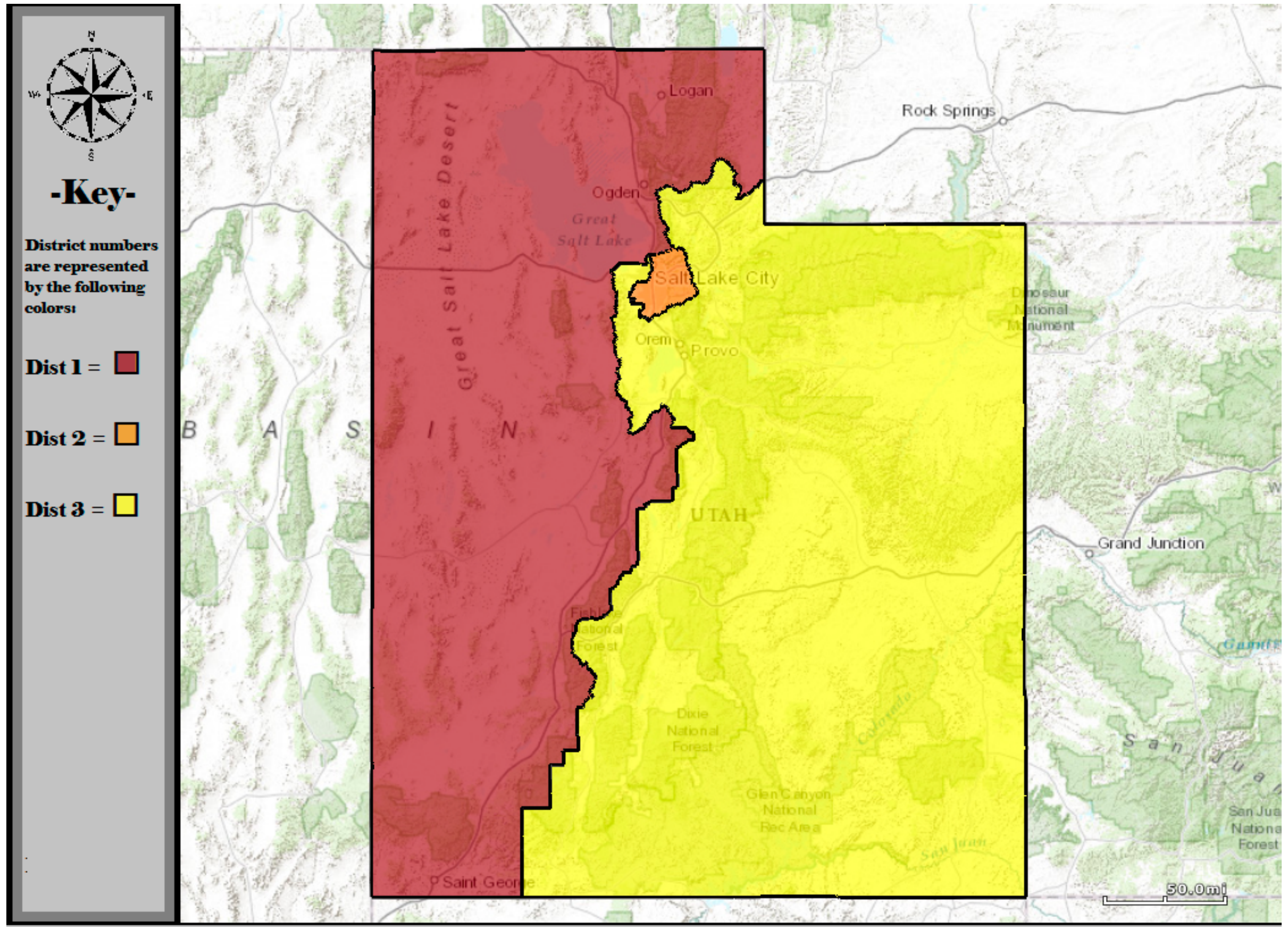
Updated Map after Court Decision



https://www.nytimes.com/interactive/2018/11/29/us/politics/north-carolina-gerrymandering.html?action=click&module=Top%20Stories&pgtype=Homepage

Congressional Districts





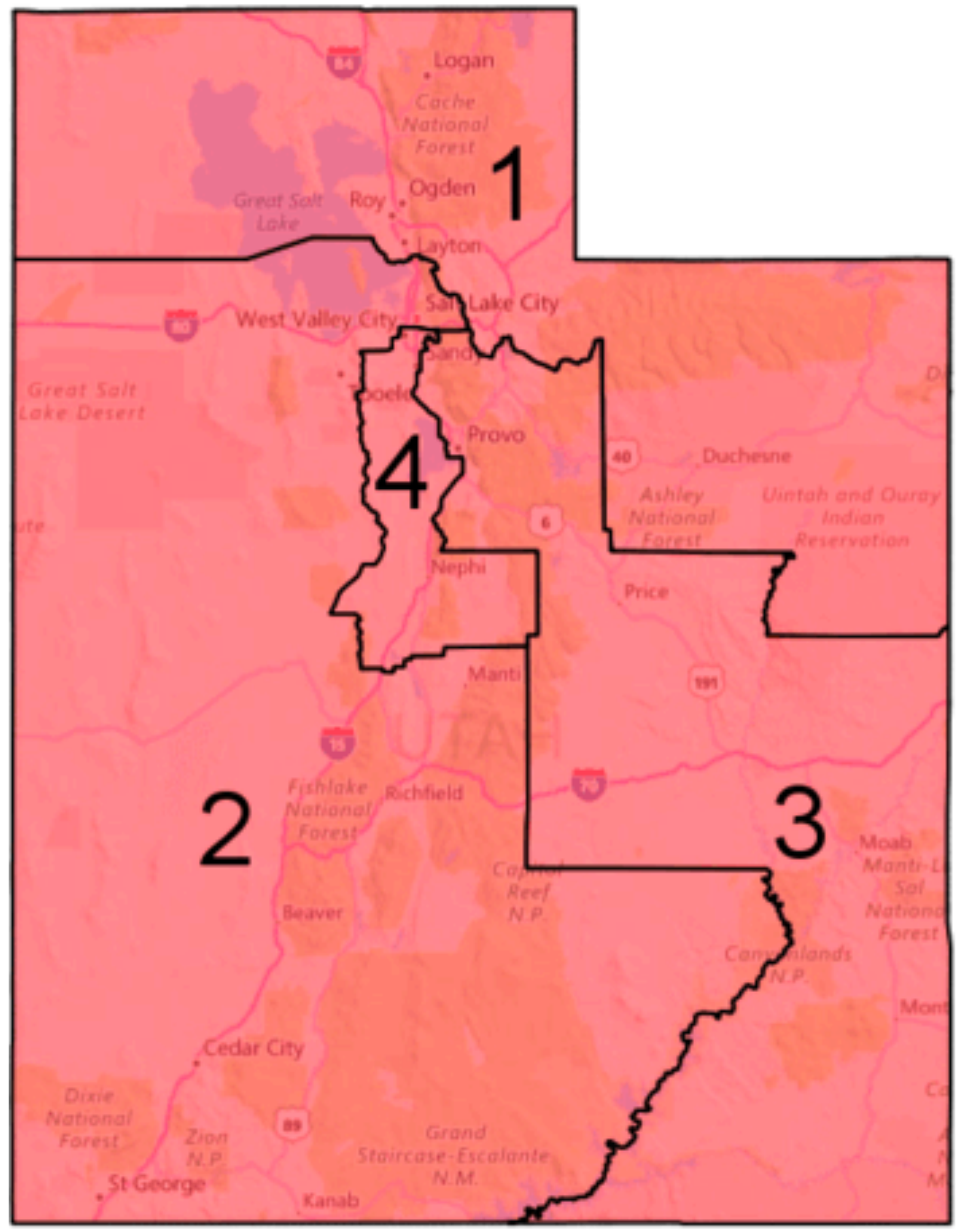
United States Congressional Districts in Utah: 1993 – 2002
Data source: http://cdmaps.polisci.ucla.edu, *Digital Boundary Definitions of United States Congressional Districts, 1789 – 2012.*

Valid till 2002

http://www.sltrib.com/opinion/1794525-155/lake-salt-republican-county-http-utah

2016 Congressional Elections

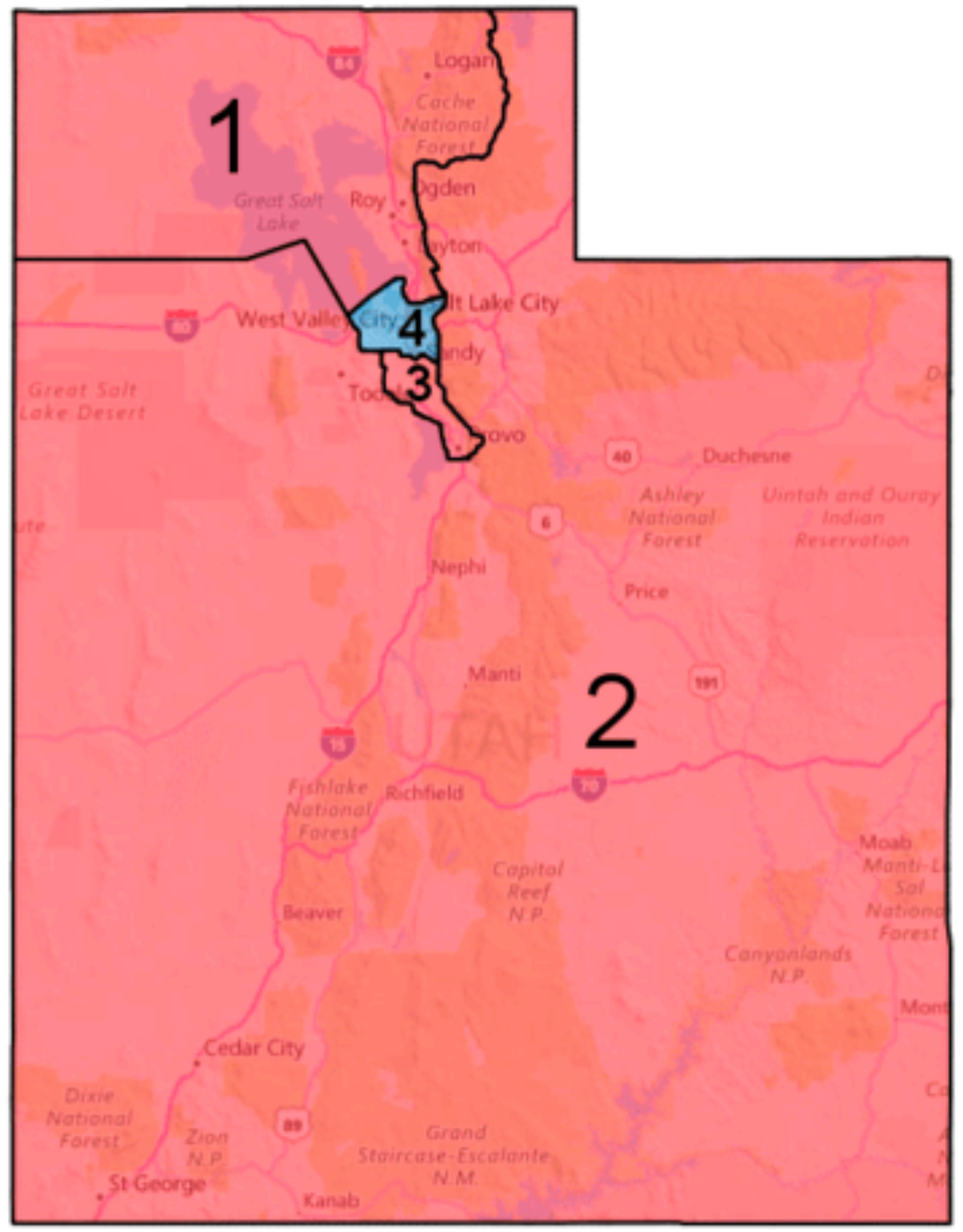
Utah's Republican Congressional Map



2016 Outcome

Republican (4)

Hypothetical Nonpartisan Map



Predicted Outcome

Democratic (1)

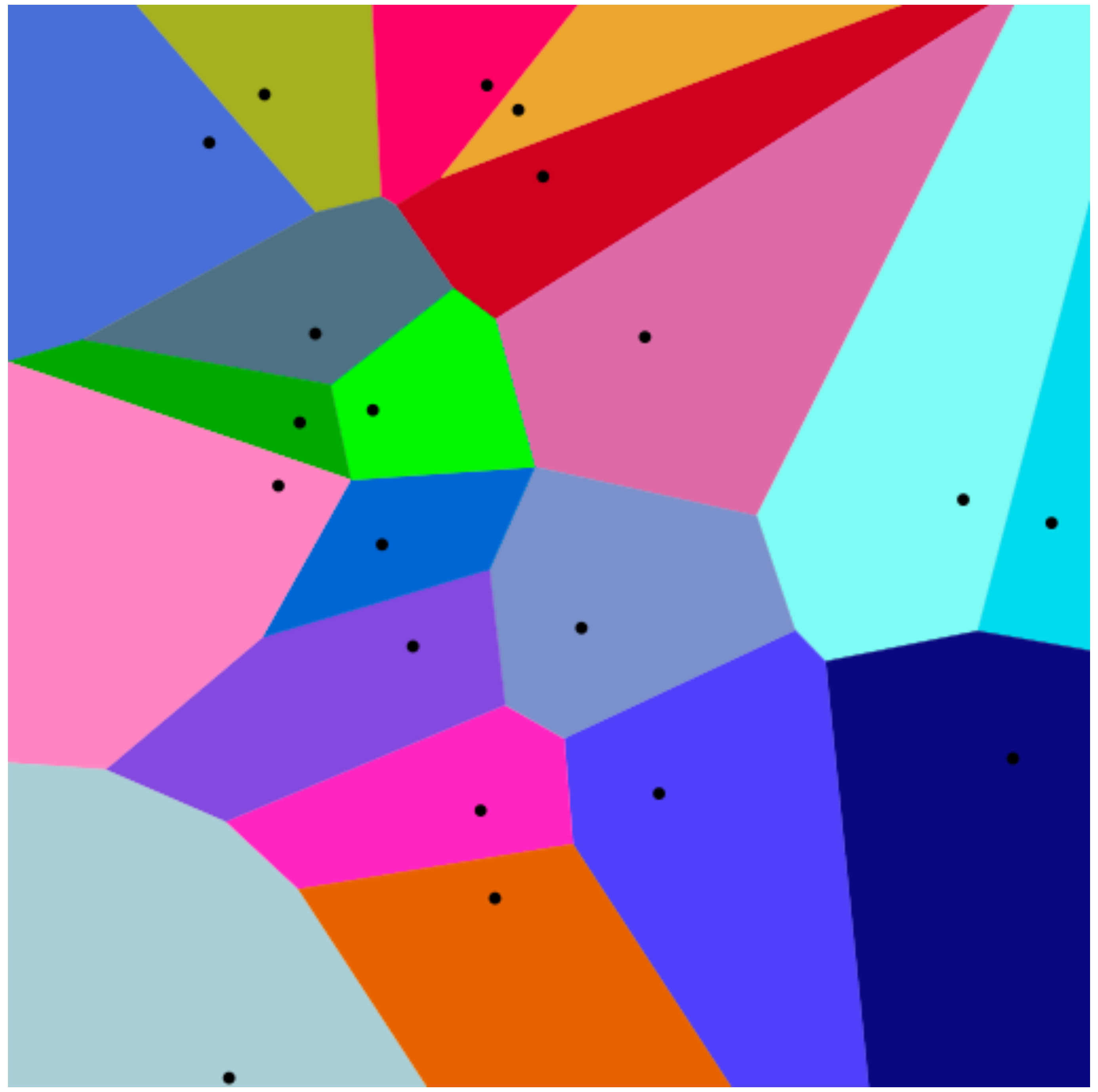
Republican (3)

DAILY KOS ELECTIONS

https://www.dailykos.com/stories/2016/12/29/1611906/-Here-s-what-Utah-might-have-looked-like-in-2016-without-congressional-gerrymandering

Voronoi Diagrams

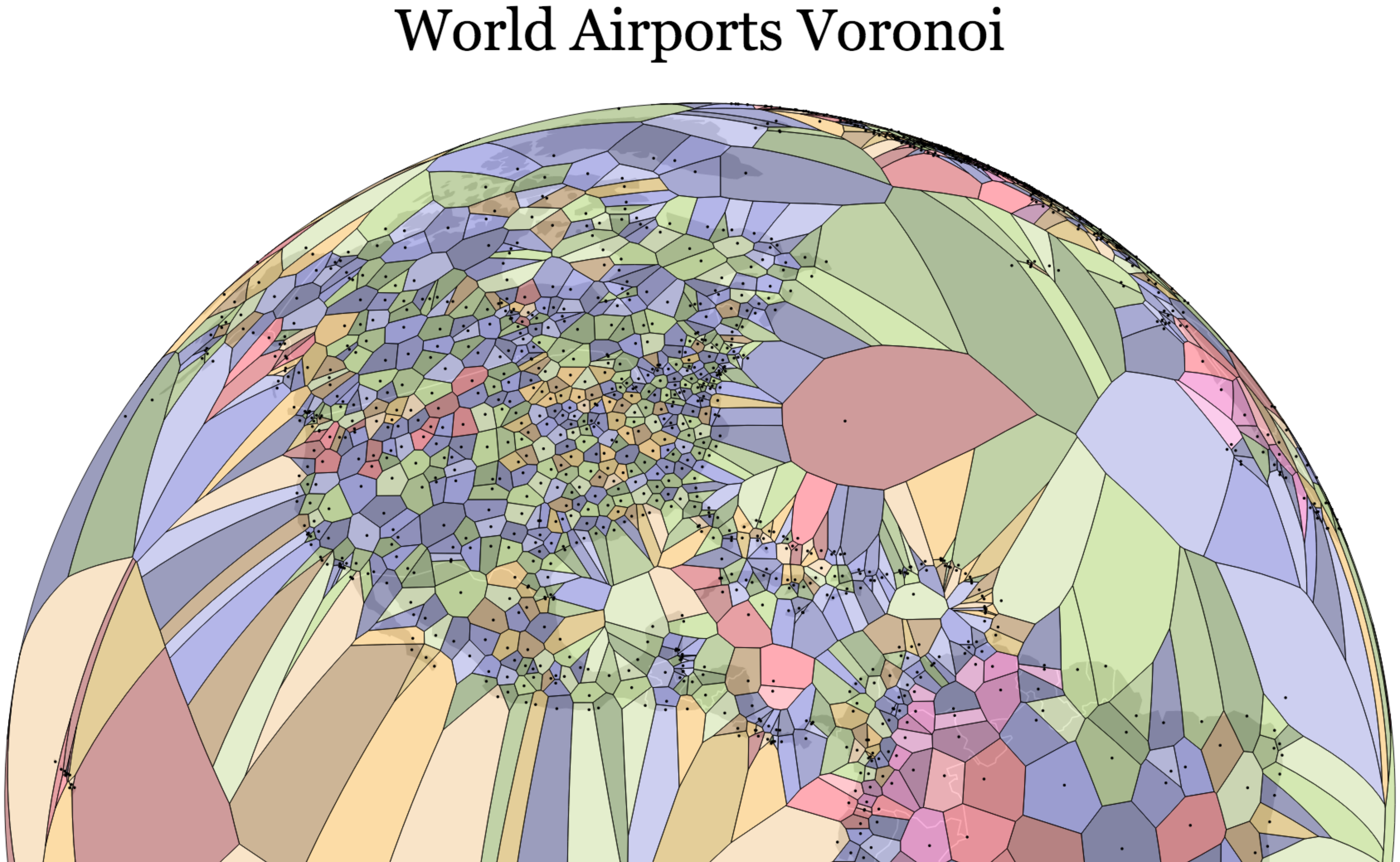
Given a set of locations, for which area is a location n closest?



D3 Voronoi Layout:

https://github.com/d3/d3-voronoi

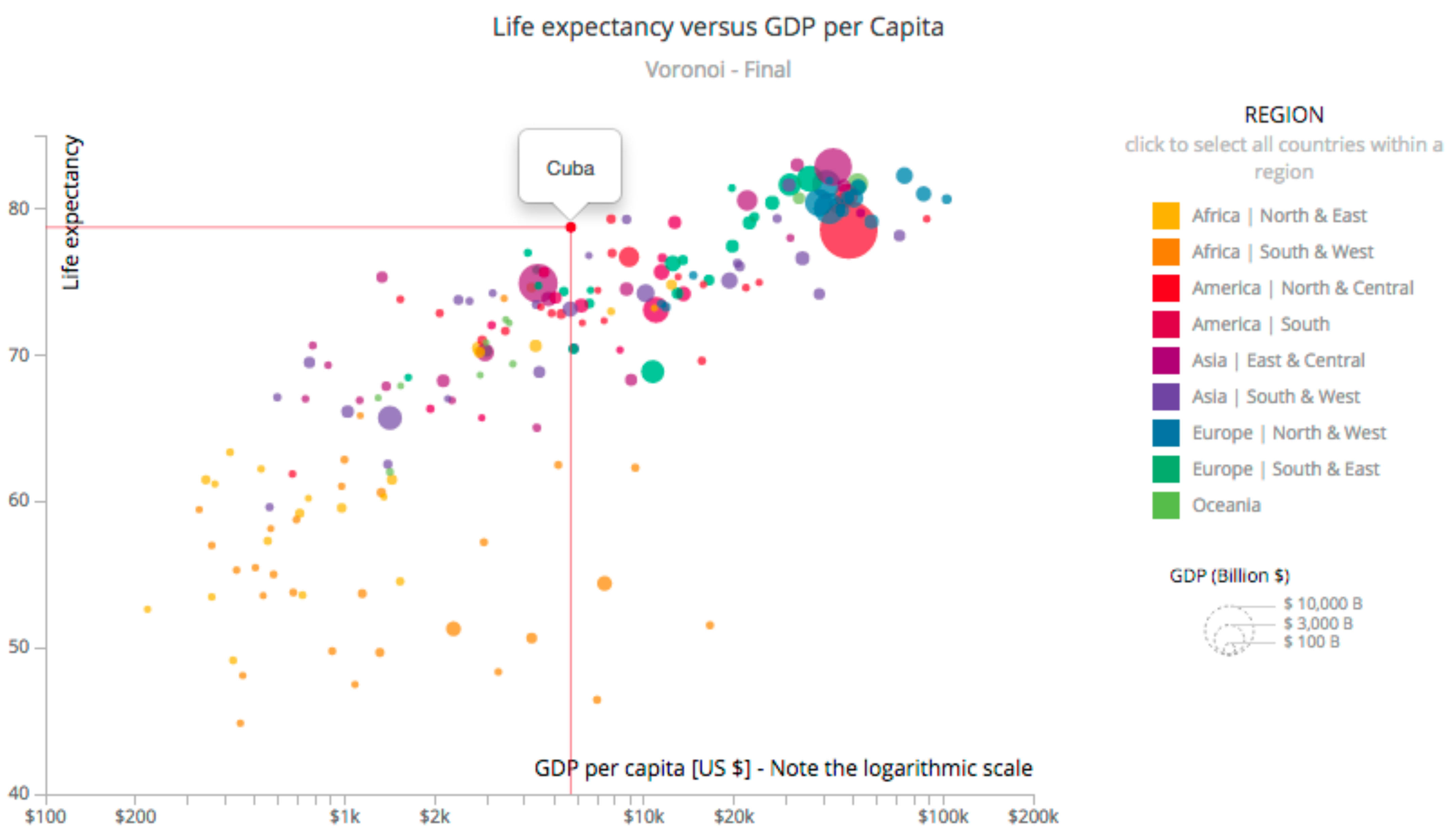
Voronoi Examples

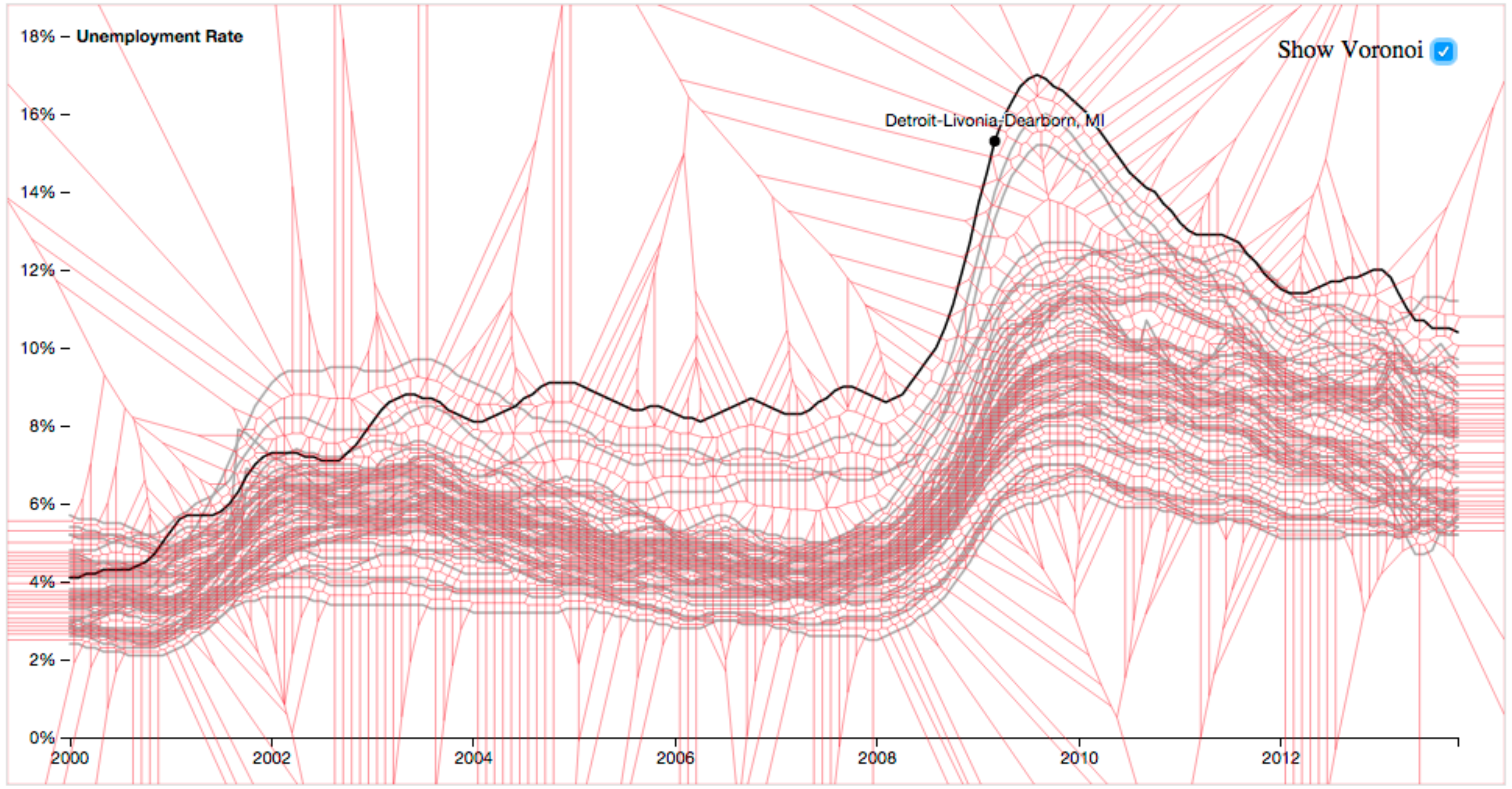


Voronoi for Interaction

Useful for interaction:
Increase size of target area to click/hover

Instead of clicking on point, hover in its region





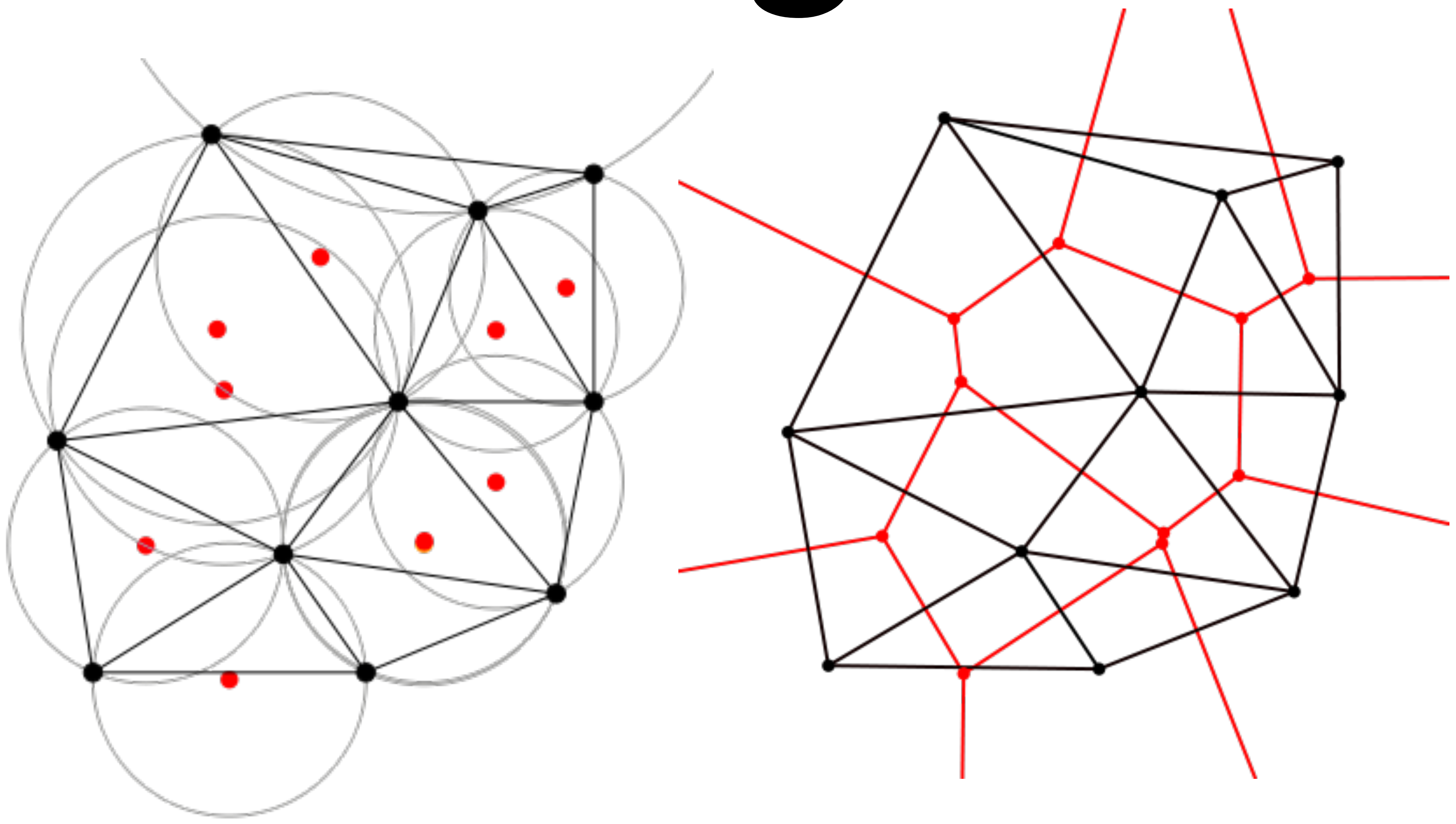
https://github.com/d3/d3-voronoi/

Constructing a Voronoi Diagram

Calculate a Delauney triangulation

- Triangulation where no other vertices are in a circle described by the vertices of a triangle

Voronoi edges are perpendicular to triangle edges.



https://en.wikipedia.org/wiki/Delaunay_triangulation

http://paulbourke.net/papers/triangulate/

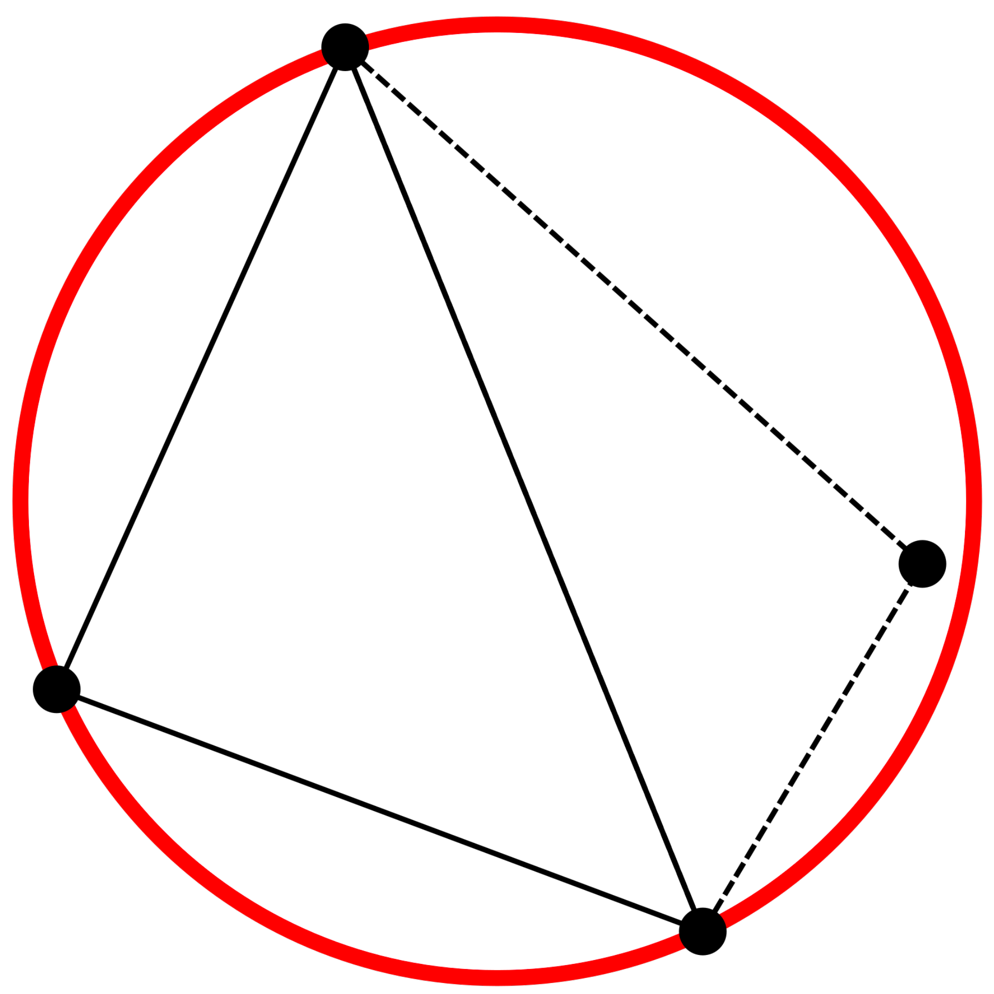
Computing a Delaunay Triangulation

Construct any triangulation

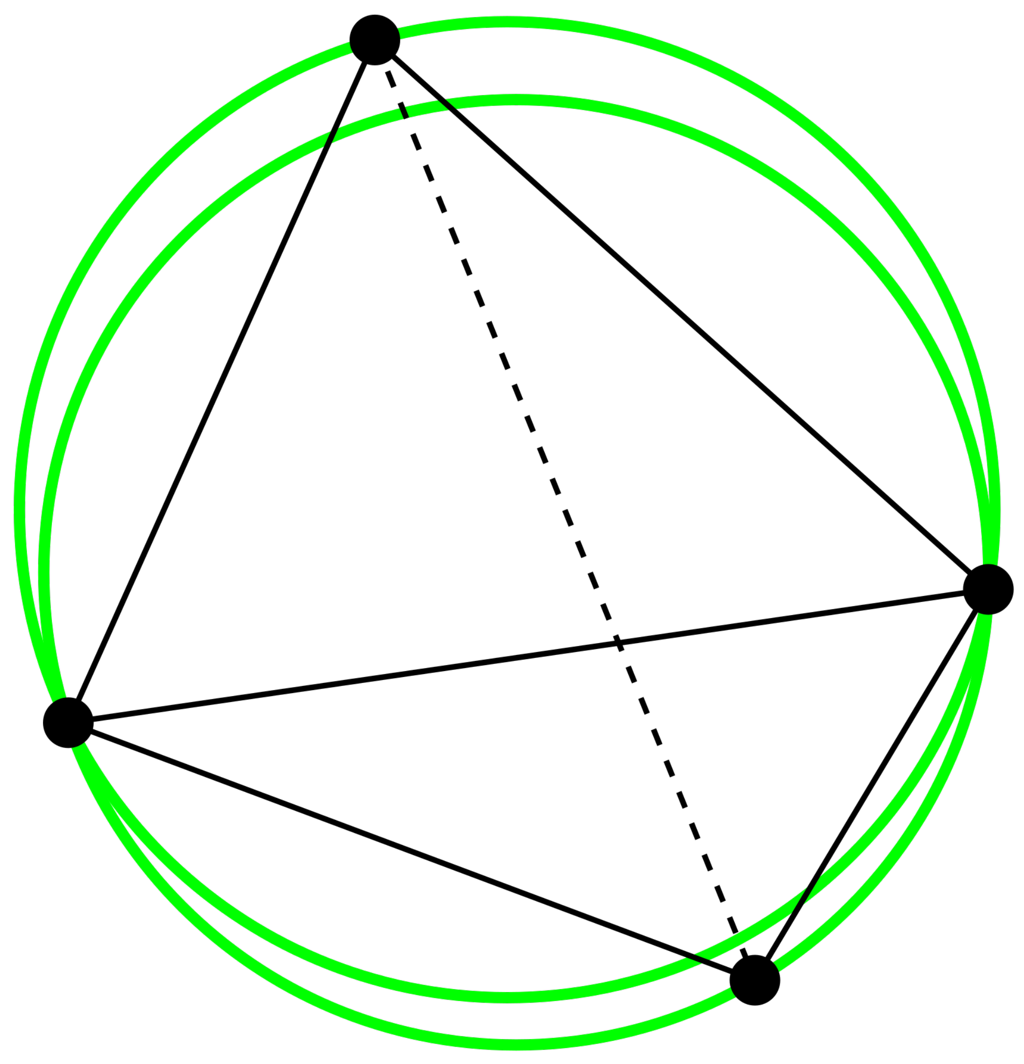
Test whether each triangle is delauny

If not, flip edge

Not a Delaunay triangle



Flipping edge produces Delaunay triangle



Clustering

Clustering

Classification of items into “similar” bins

Based on similarity measures

- Euclidean distance, Pearson correlation, ...

Partitional Algorithms

- divide data into set of bins
- # bins either manually set (e.g., k-means) or automatically determined (e.g., affinity propagation)

Hierarchical Algorithms

- Produce “similarity tree” – dendrogram

Bi-Clustering

- Clusters dimensions & records

Fuzzy clustering

- allows occurrence of elements in multiples clusters

Clustering Applications

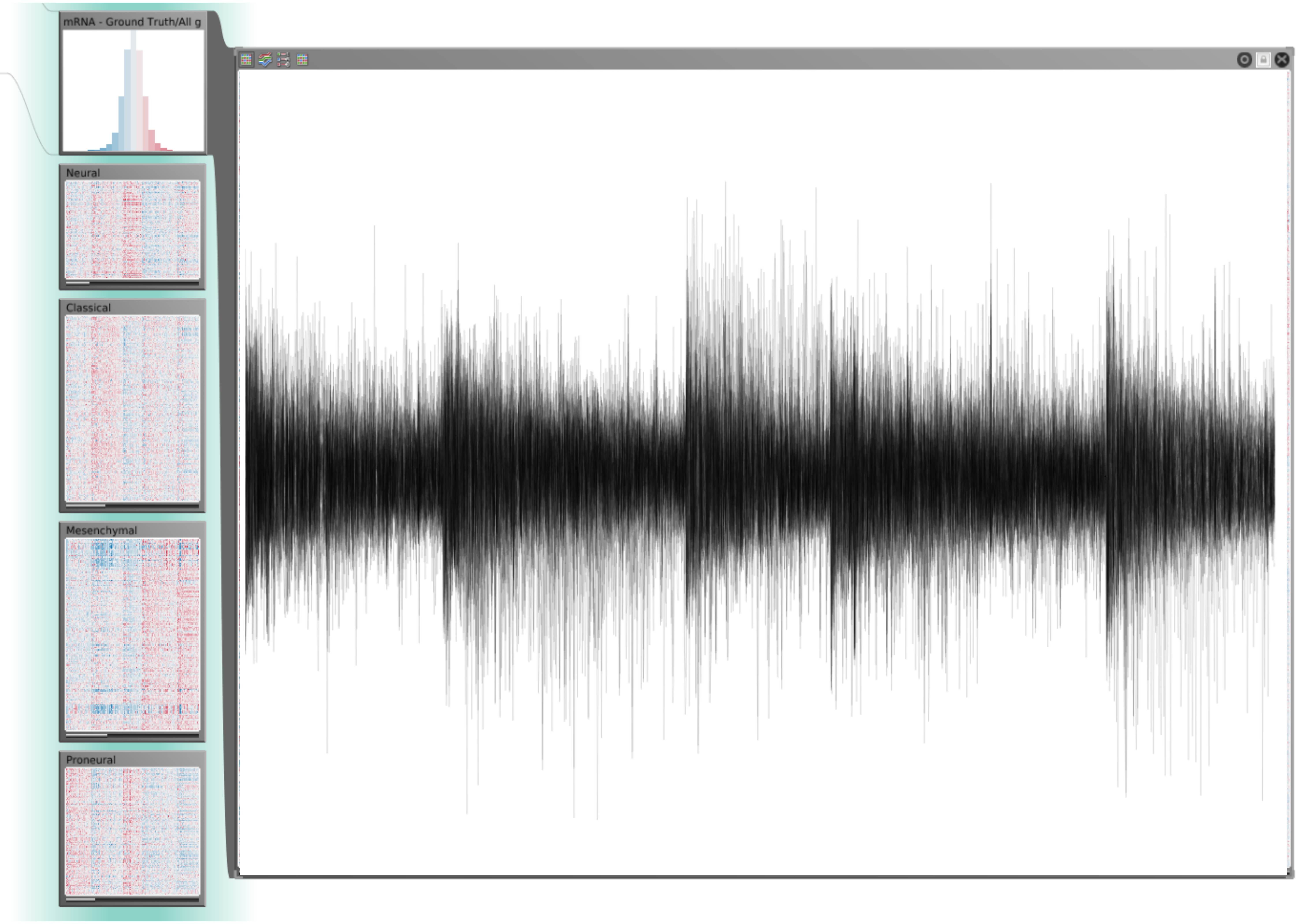
Clusters can be used to

- order (pixel based techniques)
- brush (geometric techniques)
- aggregate

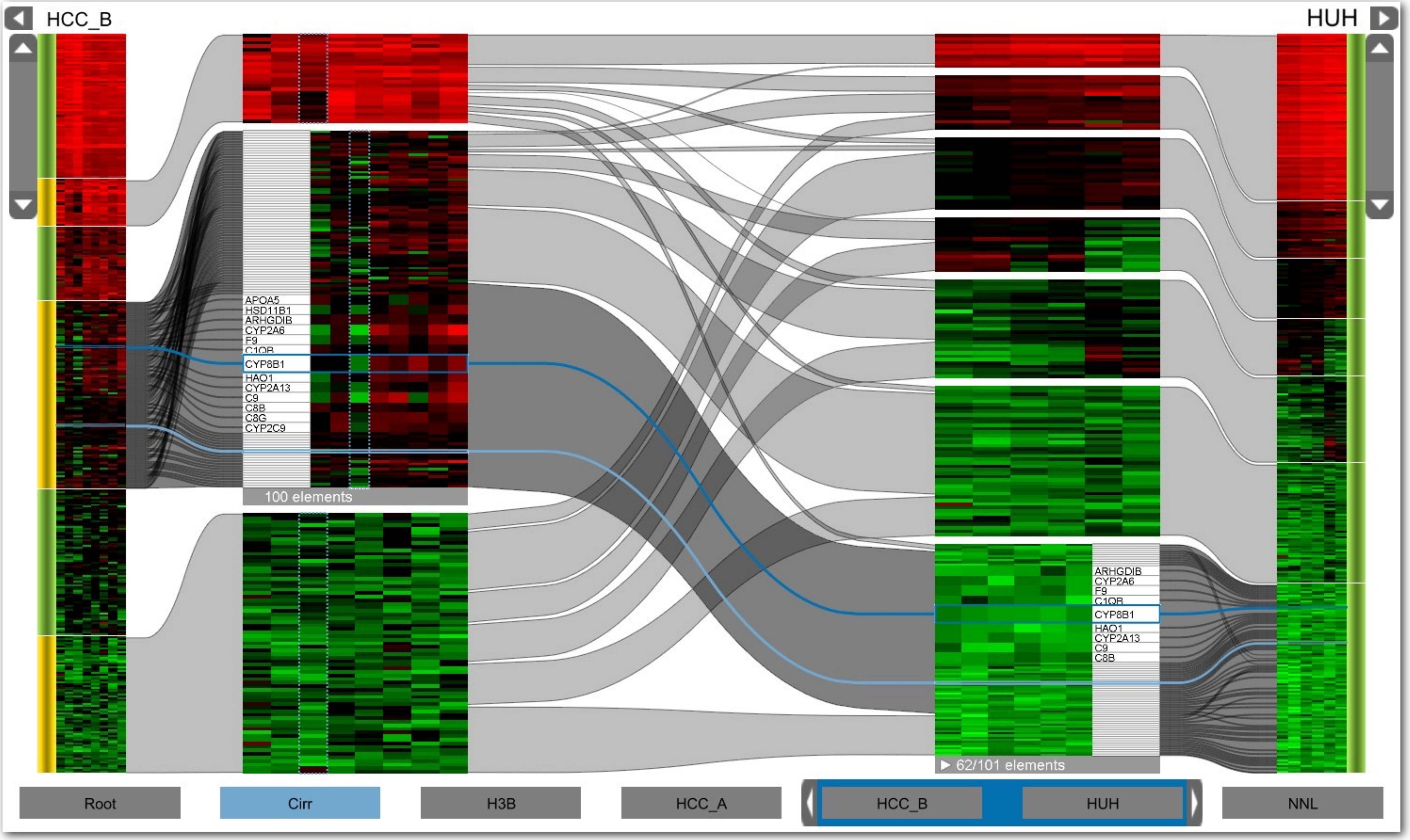
Aggregation

- cluster more homogeneous than whole dataset
- statistical measures, distributions, etc. more meaningful

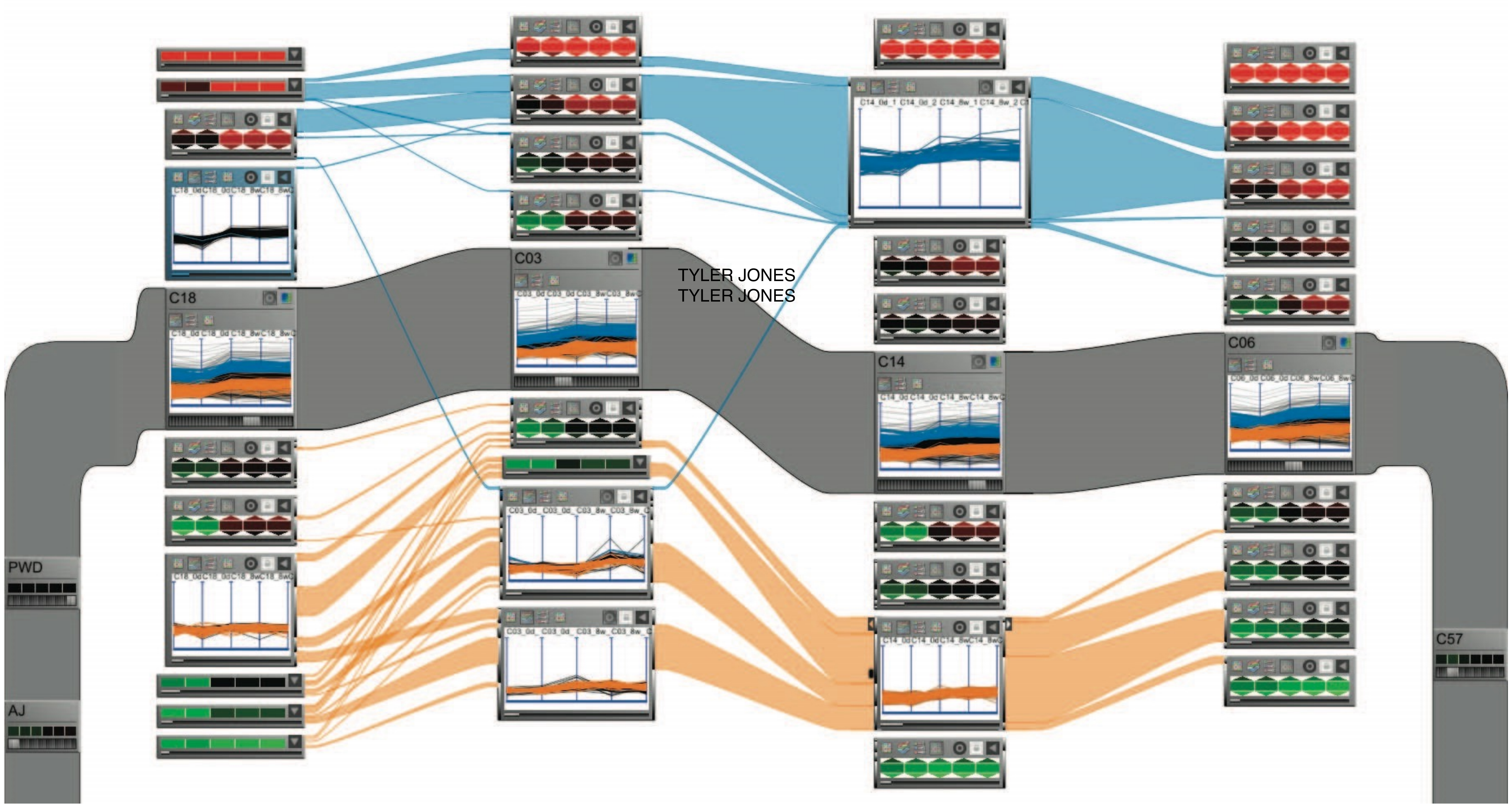
Clustered Heat Map



Cluster Comparison



Aggregation



Example: K-Means

Goal: Minimize aggregate intra-cluster distance (*inertia*)

$$\underset{C}{argmin} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

total squared distance from point to center of its cluster

for euclidian distance: this is the variance

measure of how internally coherent clusters are

Lloyd's Algorithm

Input: set of records $x_1 \ldots x_n$, and k (# of clusters)

Pick k starting points as centroids $c_1 \ldots c_k$

While not converged:

1. for each point x_i find closest centroid c_j
   - for every c_j calculate distance $D(x_i, c_j)$
   - assign x_i to cluster j defined by smallest distance
2. for each cluster j, compute a new centroid c_j
   by calculating the average of all x_i assigned to cluster j

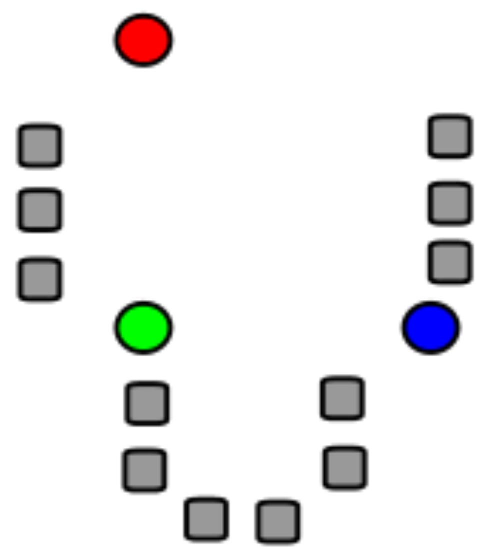
Repeat until convergence, e.g.,

no point has changed cluster

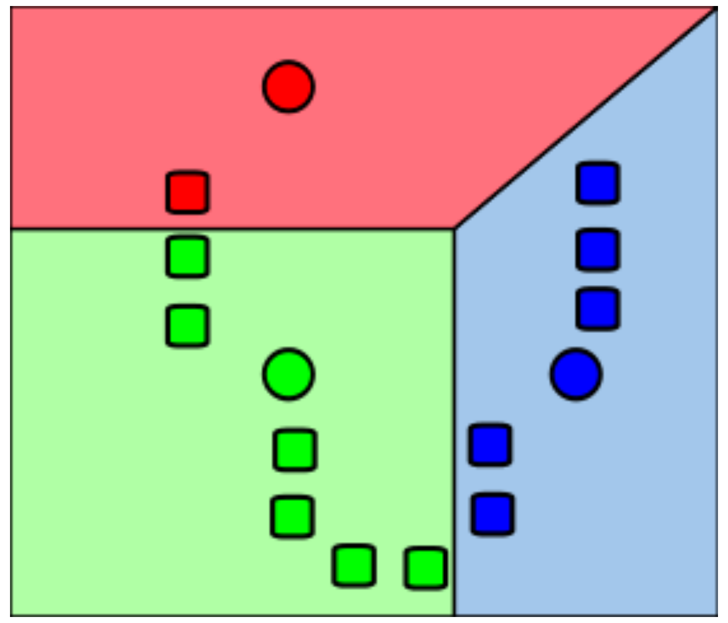
distance between old and new centroid below threshold

number of max iterations reached

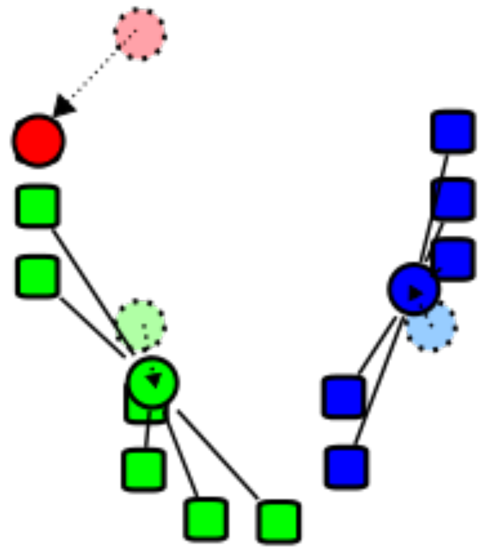
1. Initialization



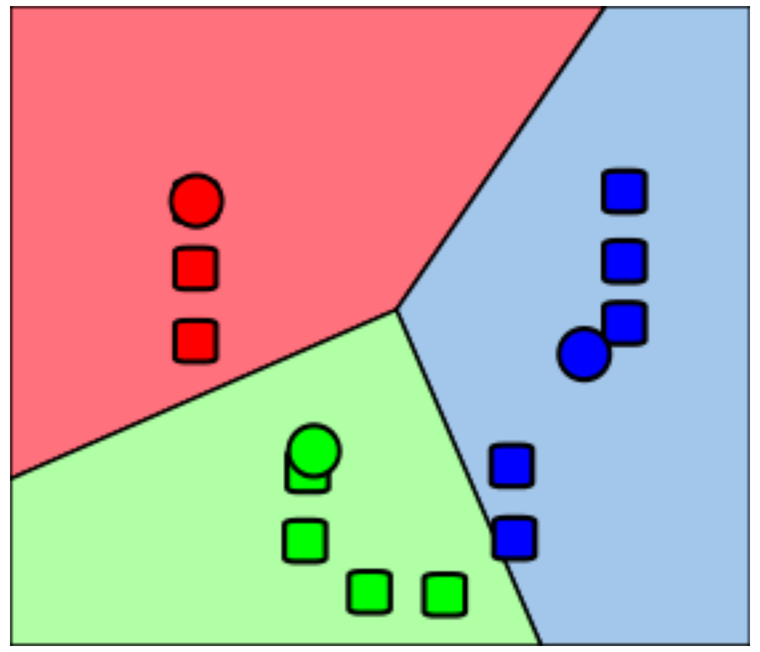
2. Assign Clusters



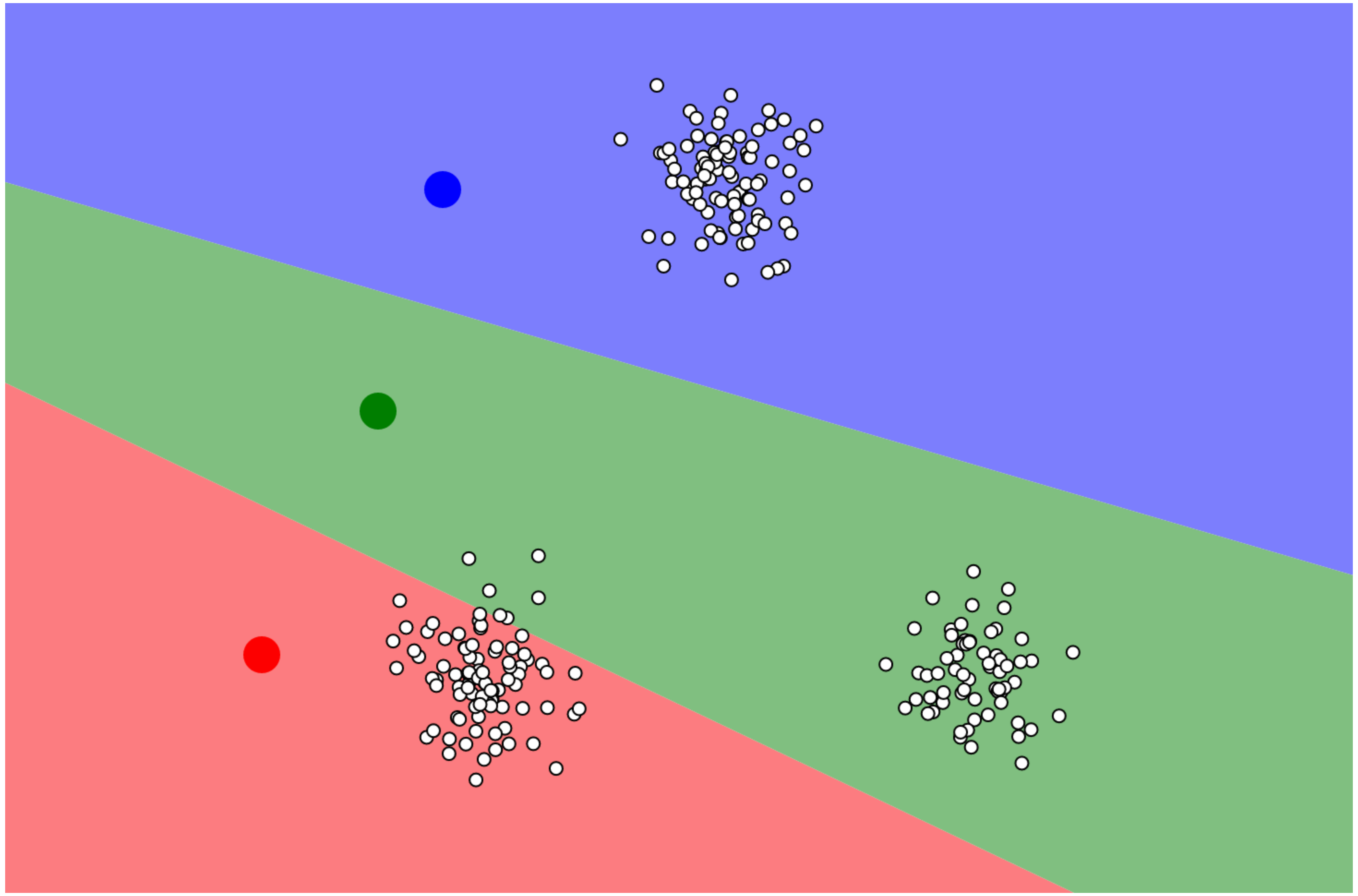
3. Update Centroids



4. Assign Clusters

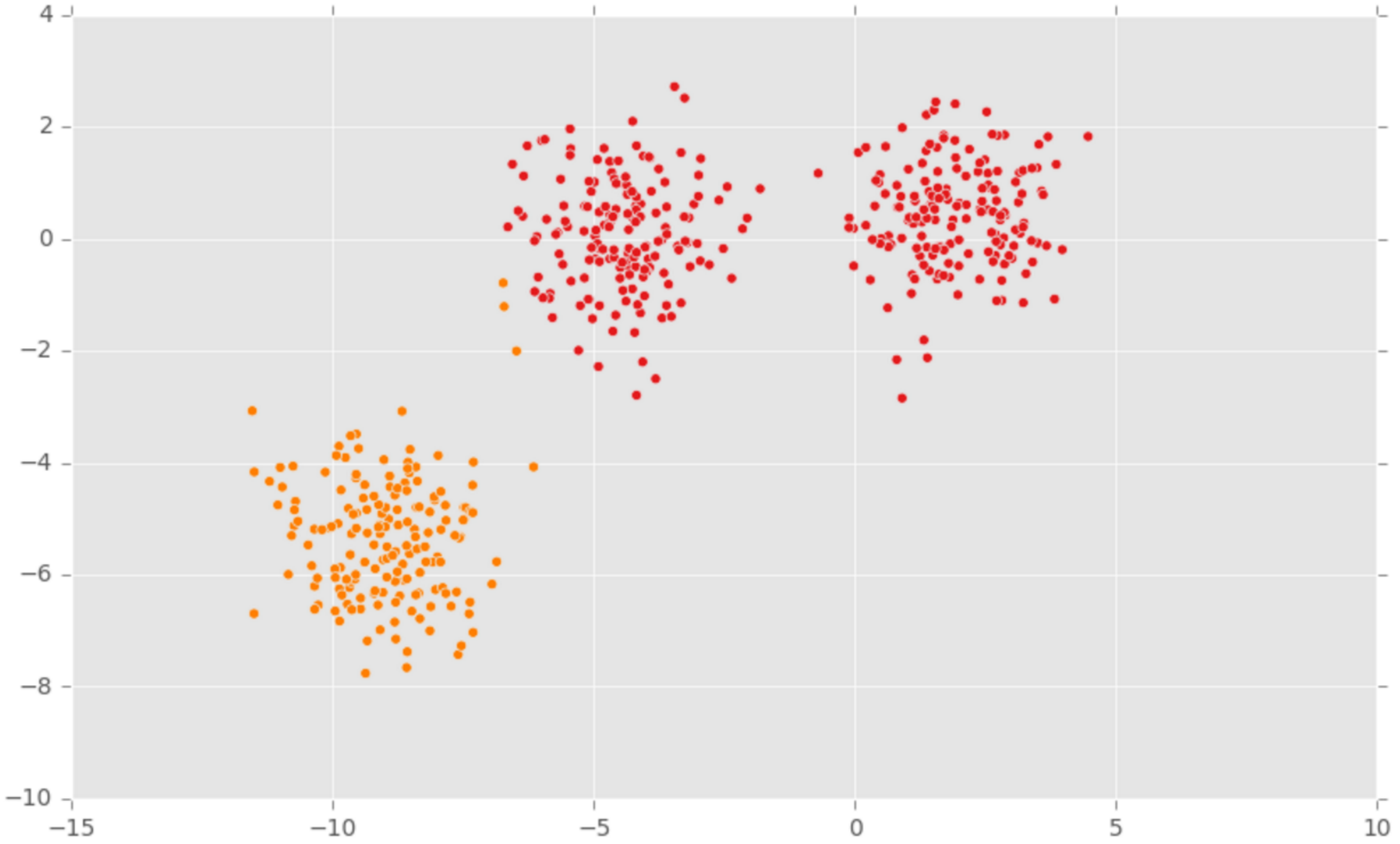


Illustrated



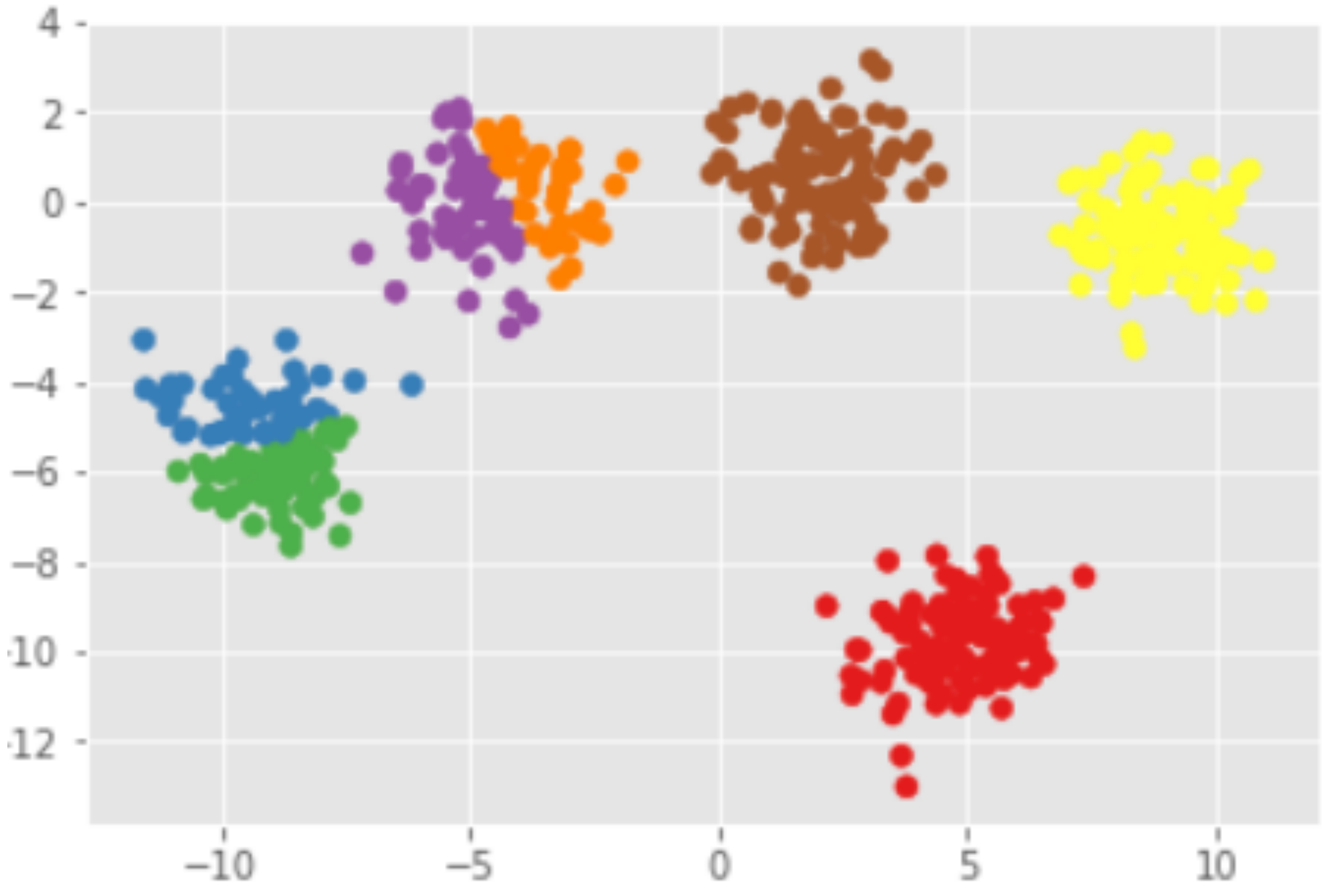
https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

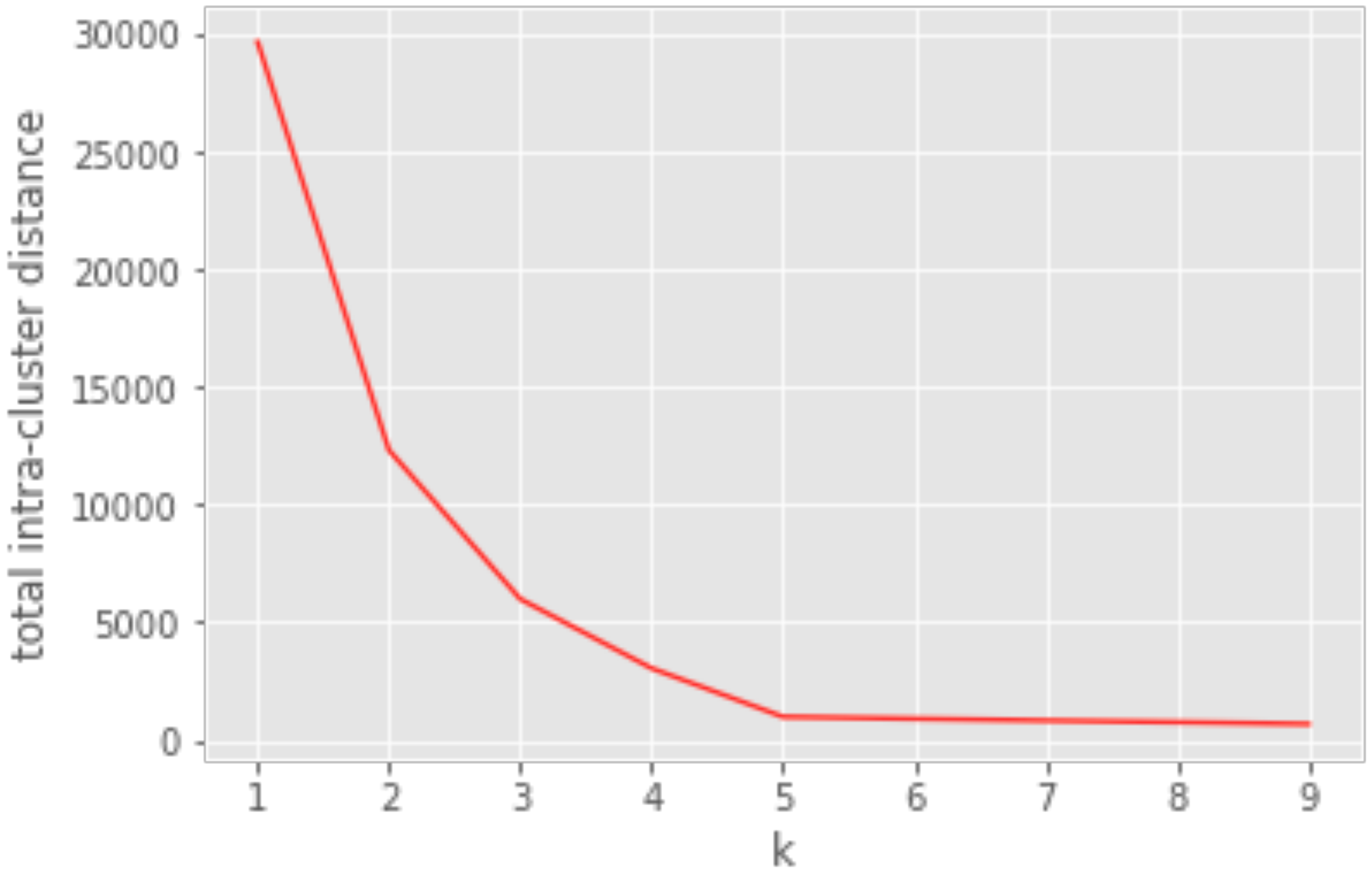
Choosing K, Initializing



Initializing: Farthest Point Strategy
Choosing K: looking for drop-off in Intra-Cluster Distance Reduction

Evaluating Intra-Cluster Distance





Properties

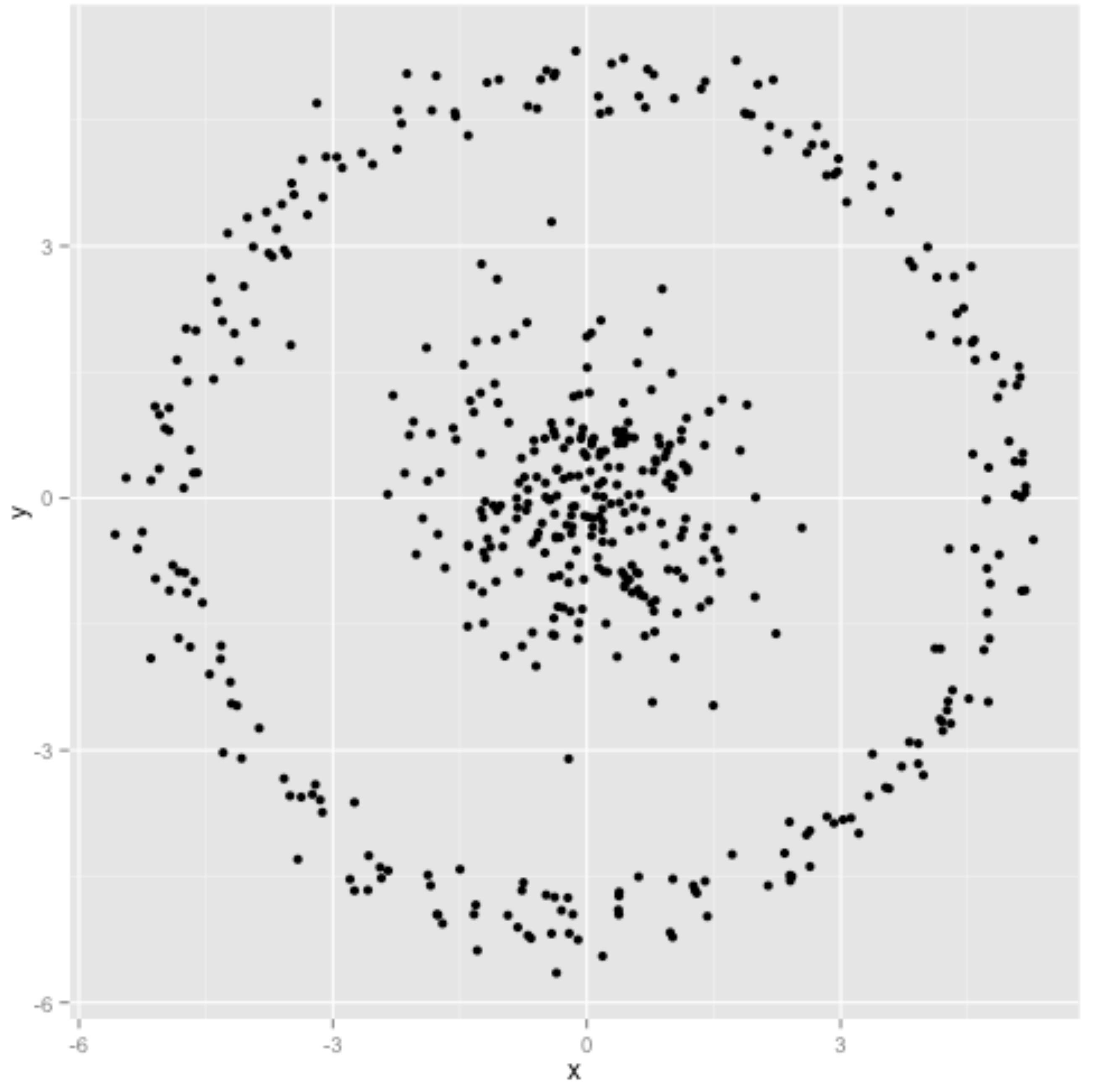
Lloyds algorithm doesn’t find a global optimum

Instead it finds a local optimum

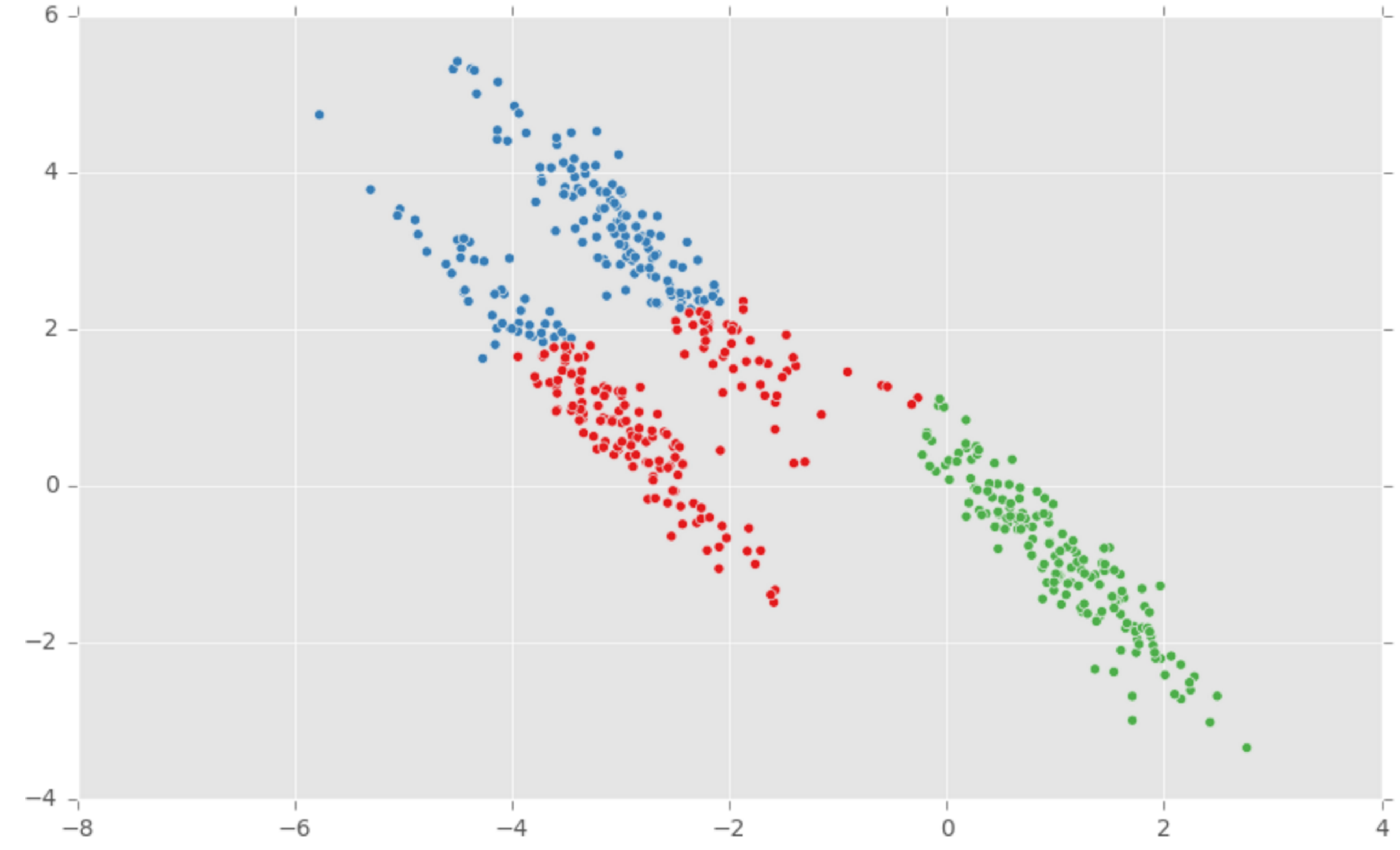
It is very fast:

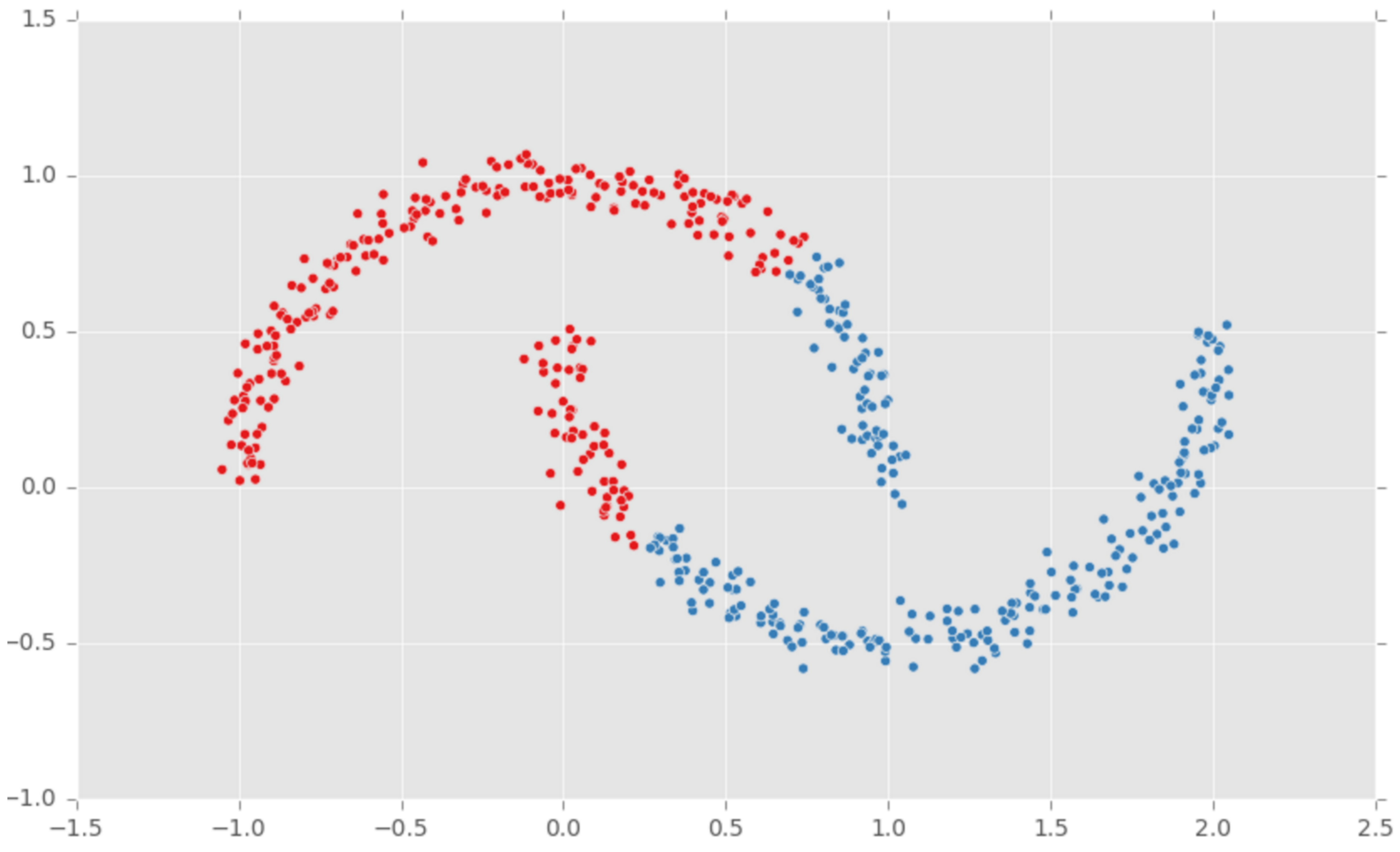
- common to run multiple times and pick the solution with the minimum inertia

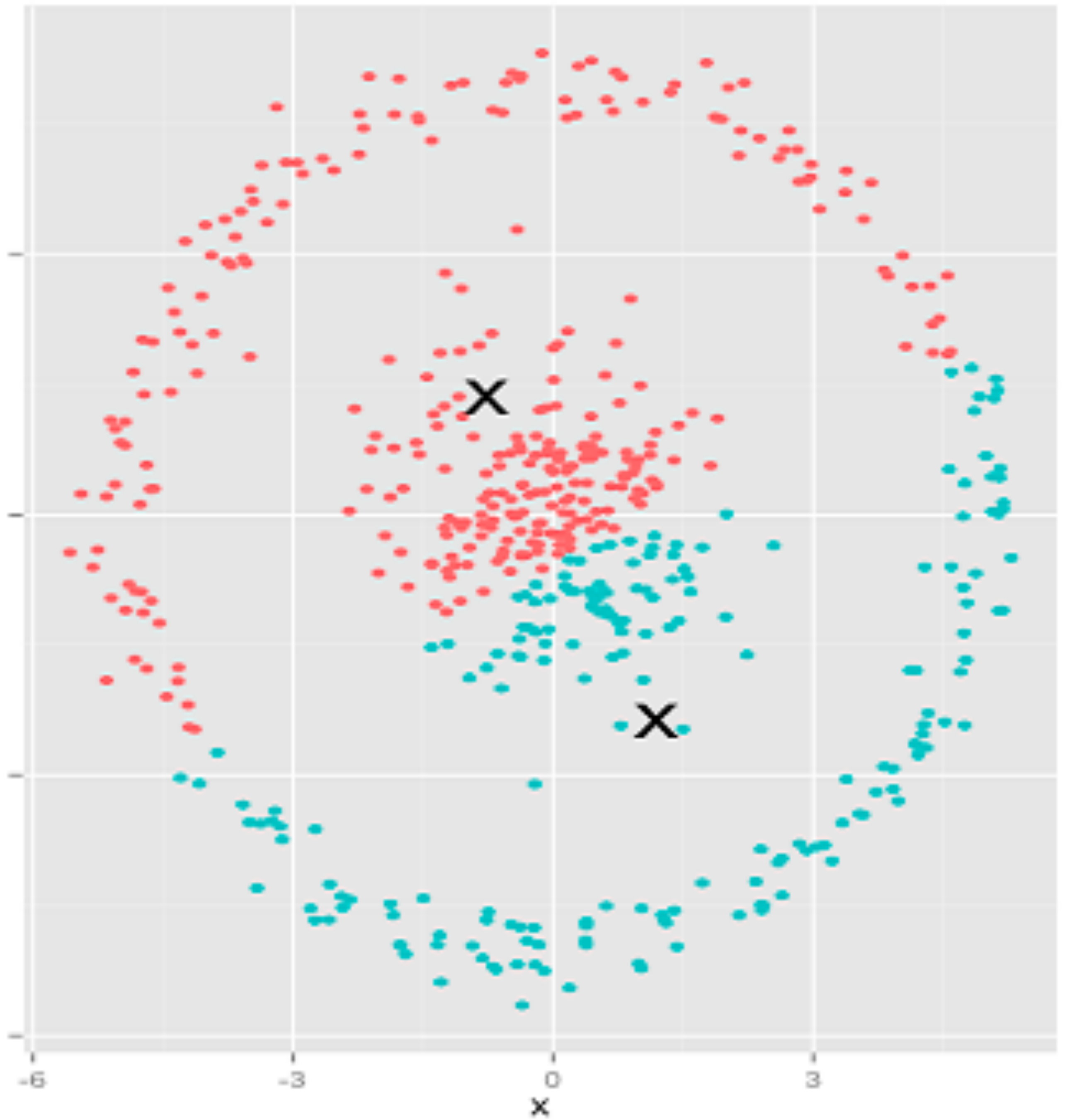
K-Means Properties



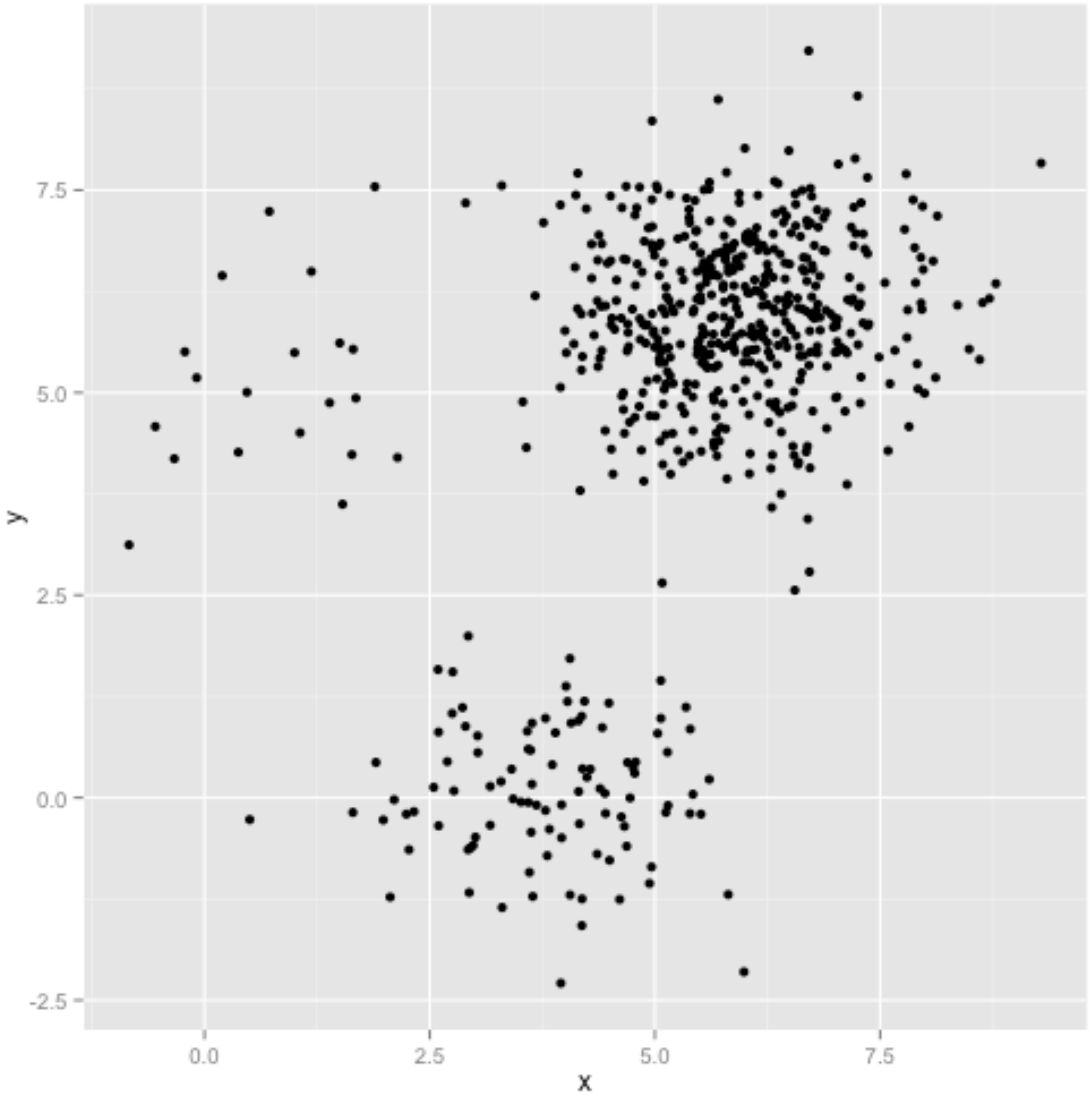
Assumptions about data: roughly "circular" clusters of equal size







K-Means Unequal Cluster Size



http://stats.stackexchange.com/questions/133656/how-to-understand-the-drawbacks-of-k-means

DBScan

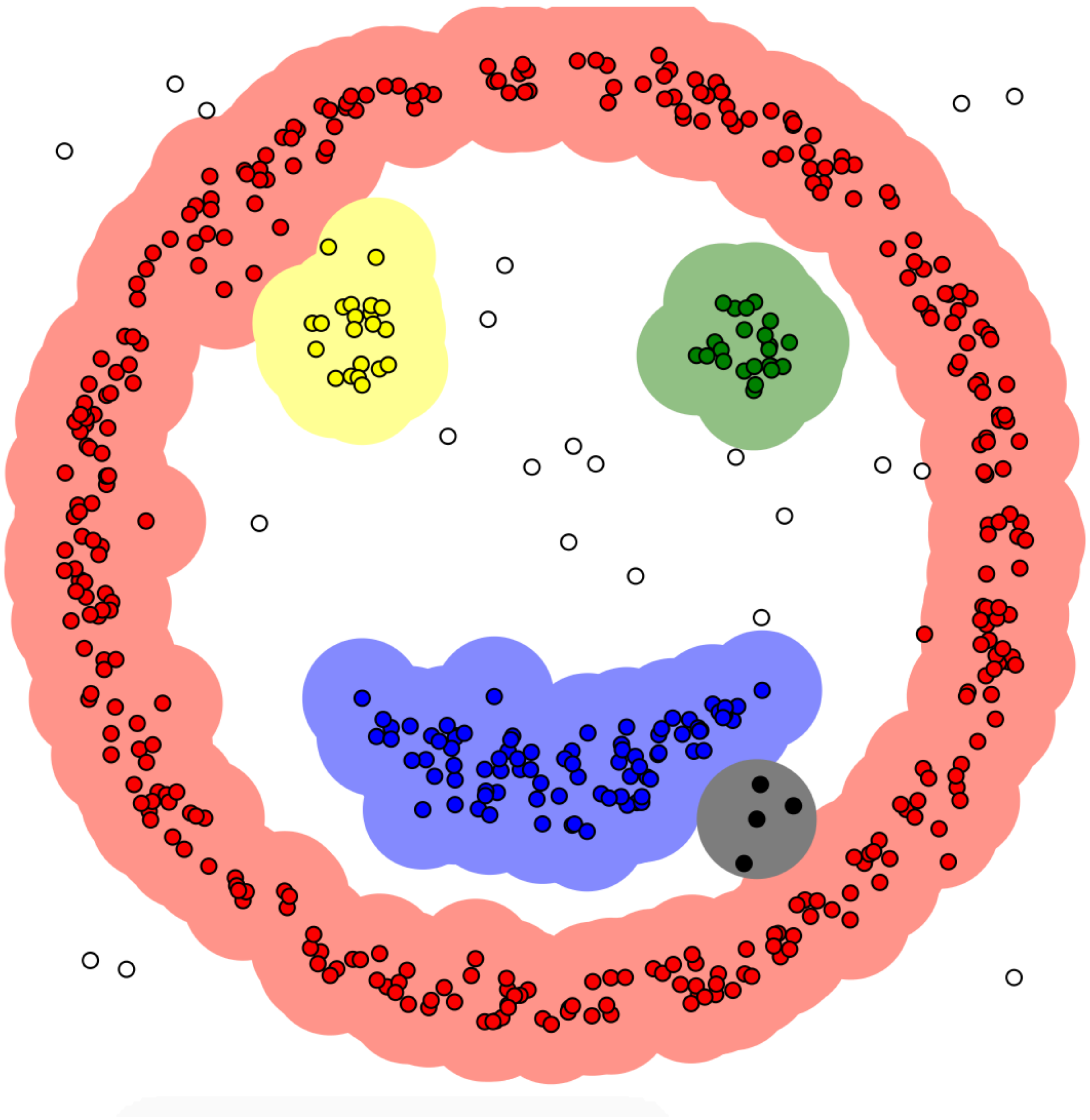
Density-based spatial clustering of applications with noise

Idea: Clusters are dense groups

if point belongs to a cluster, it should be near to lots of other points in that cluster.

Parameters:

- Epsilon: if new point distance to closest point in cluster is < epsilon, add to cluster
- Min points: what's the smallest cluster (outliers)



https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Hierarchical Clustering

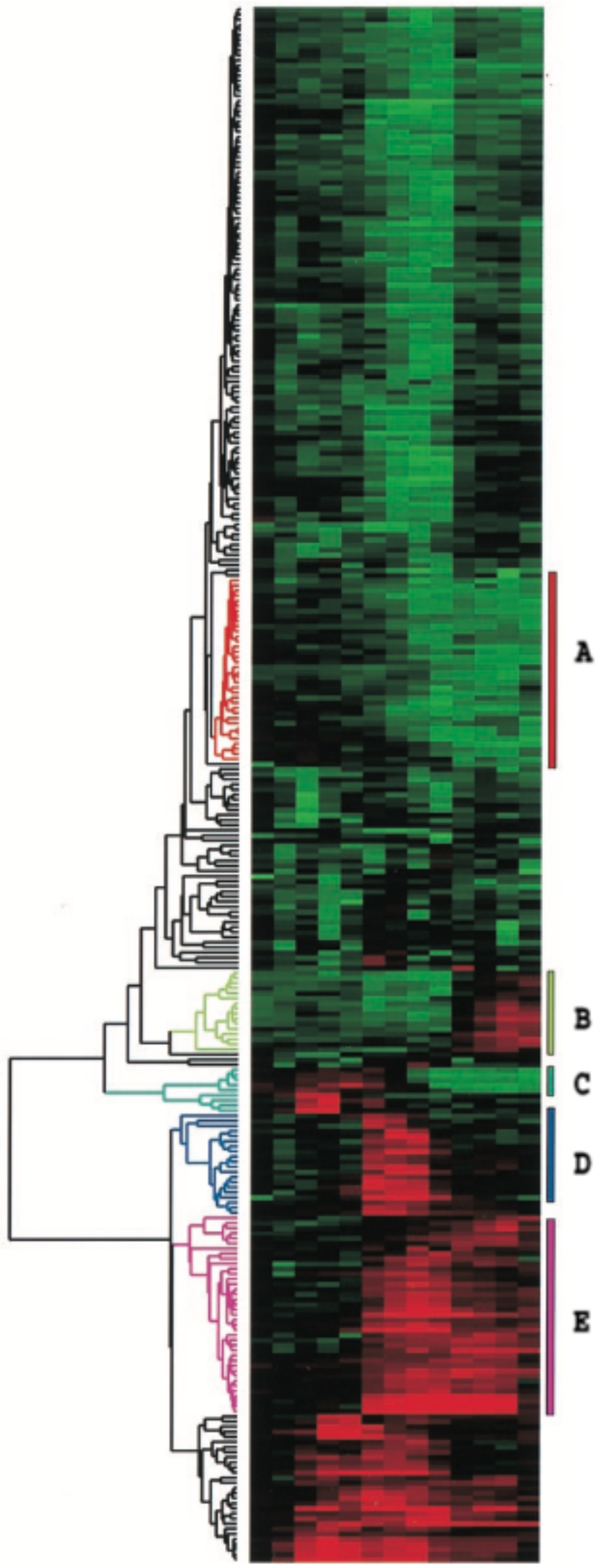
Two types:

agglomerative clustering

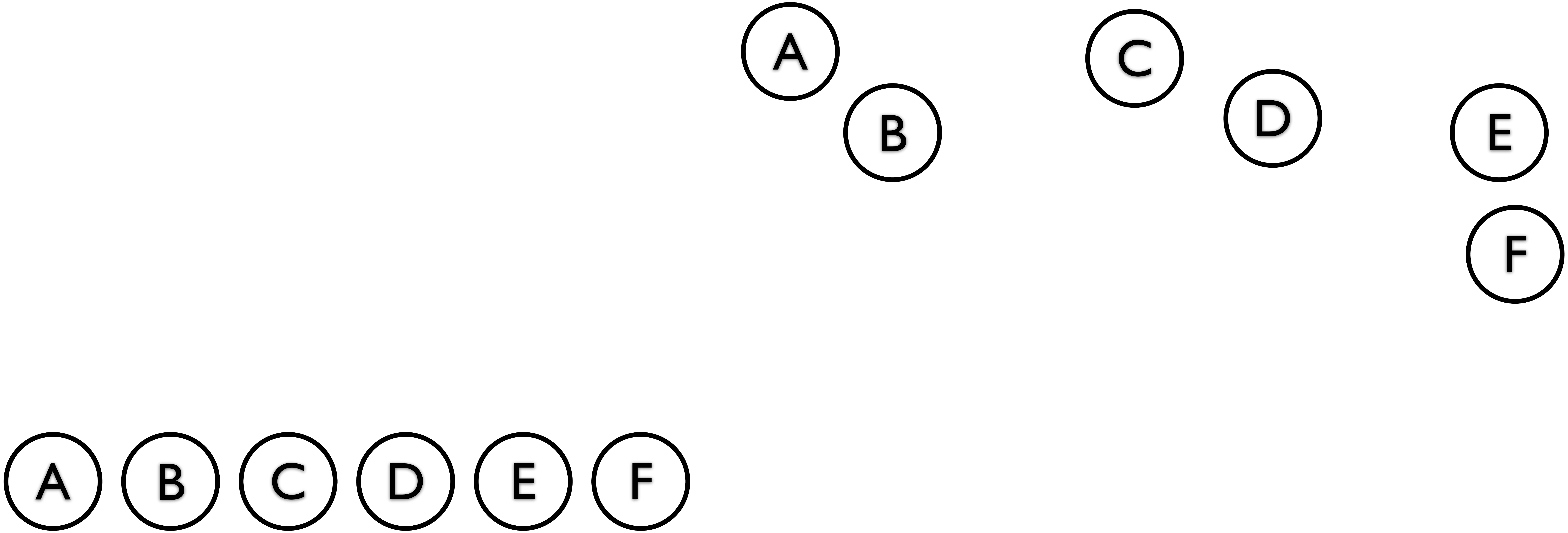
start with each node as a cluster and merge

divisive clustering

start with one cluster, and split



Agglomerative Clustering Idea



https://youtu.be/XJ3194AmH40?t=4m29s

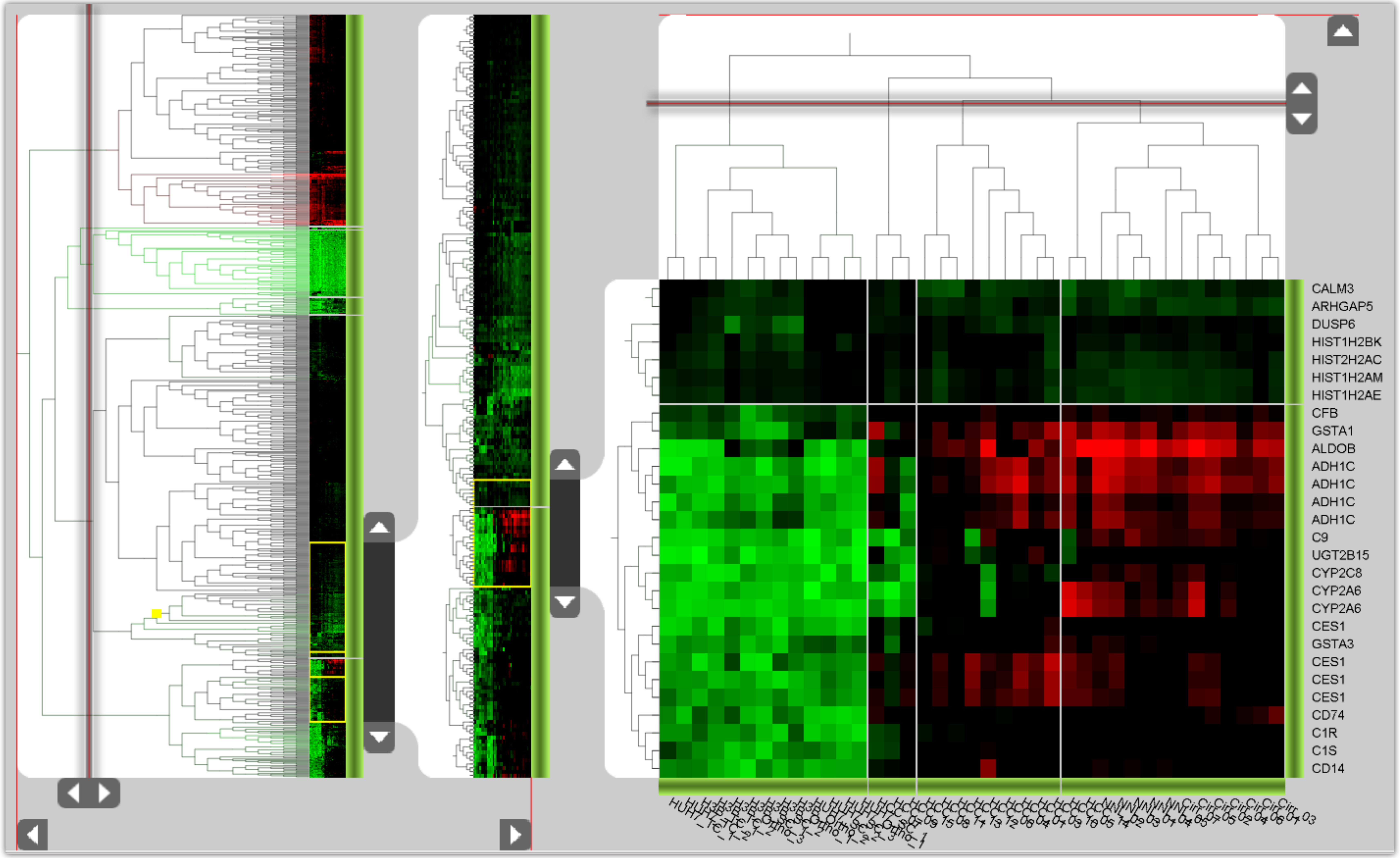
Linkage Criteria

How do you define similarity between two clusters to be merged (A and B)?

- maximum linkage distance: two elements that are apart the furthest
- use minimum linkage distance: the two closest elements
- use average linkage distance
- use centroid distance

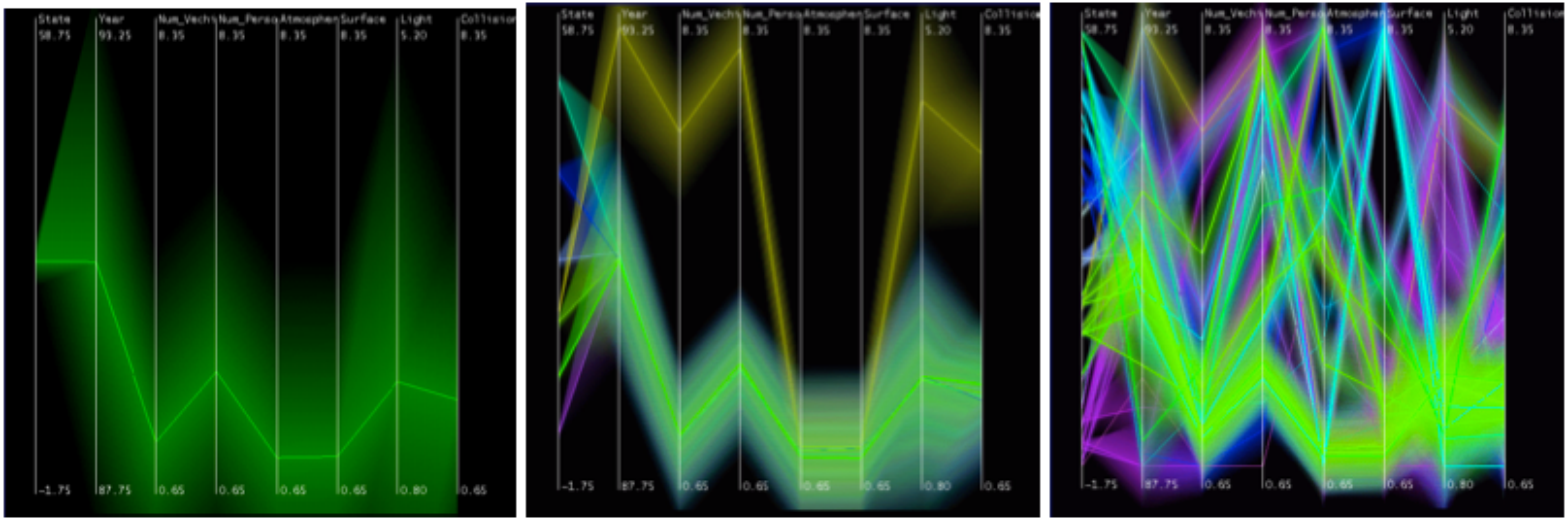
Names	Formula
Maximum or complete-linkage clustering	$\max\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Minimum or single-linkage clustering	$\min\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Mean or average linkage clustering, or UPGMA	$\frac{1}{\lvert A\rvert \lvert B\rvert} \sum_{a\in A} \sum_{b\in B} d(a,b).$
Centroid linkage clustering, or UPGMC	$\lVert c_s - c_t \rVert$ where c_s and c_t are the centroids of clusters s and t, respectively.

F+C Approach, with Dendrograms



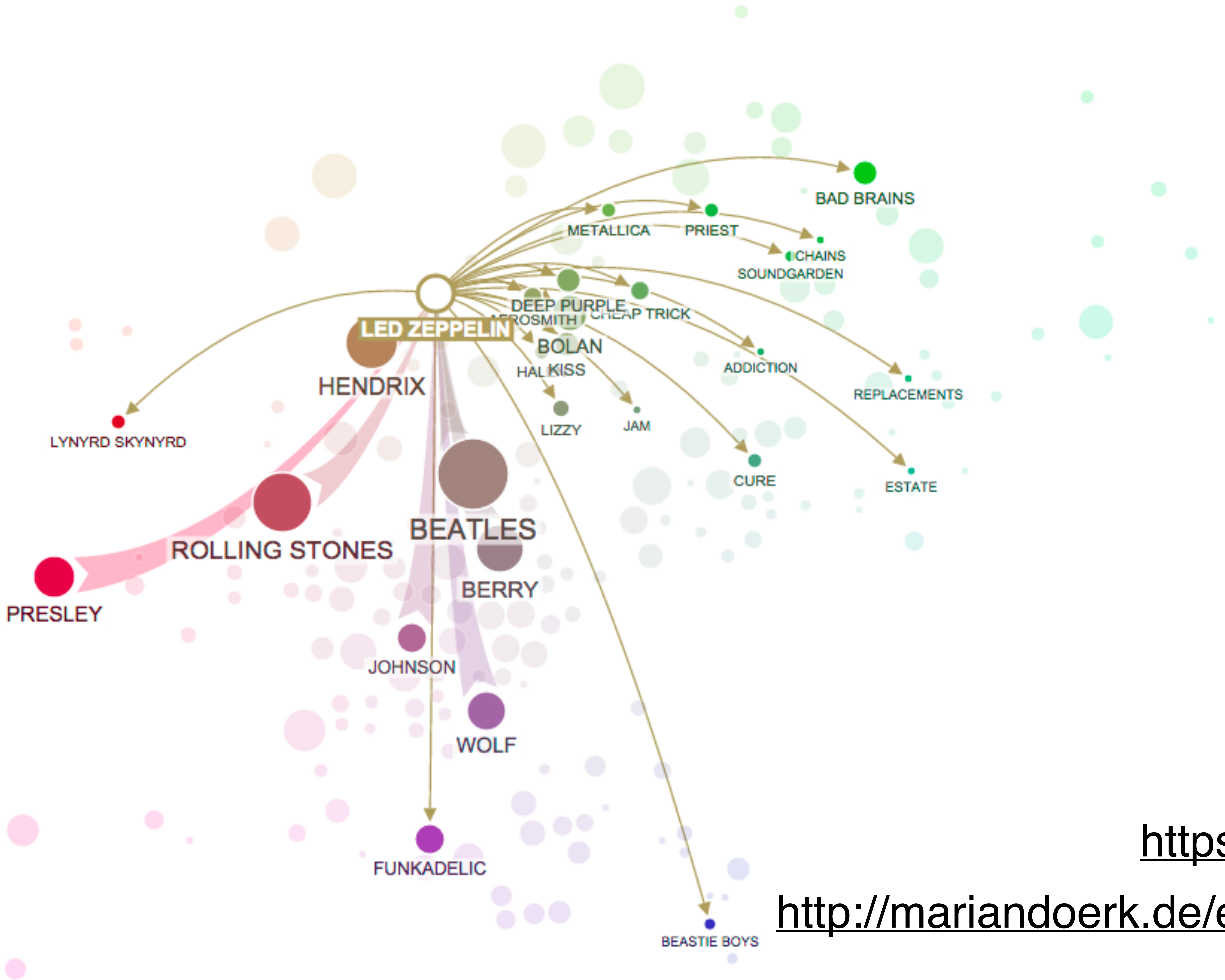
[Lex, PacificVis 2010]

Hierarchical Parallel Coordinates



Fua 1999

Design Critique



https://goo.gl/IDRXDl

http://mariandoerk.de/edgemaps/demo/

Dimensionality Reduction

Dimensionality Reduction

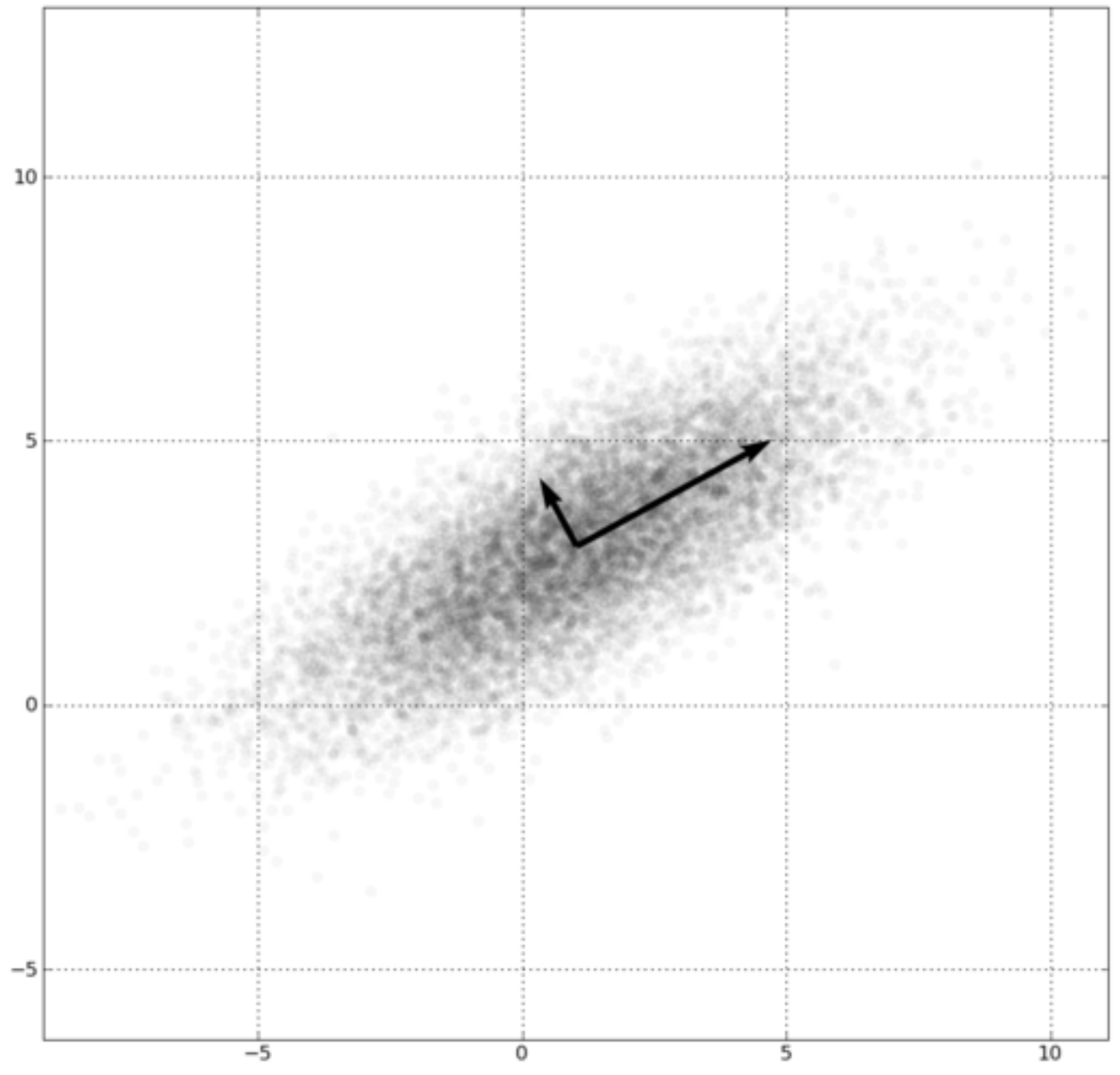
Reduce high dimensional to lower dimensional space

Preserve as much of variation as possible

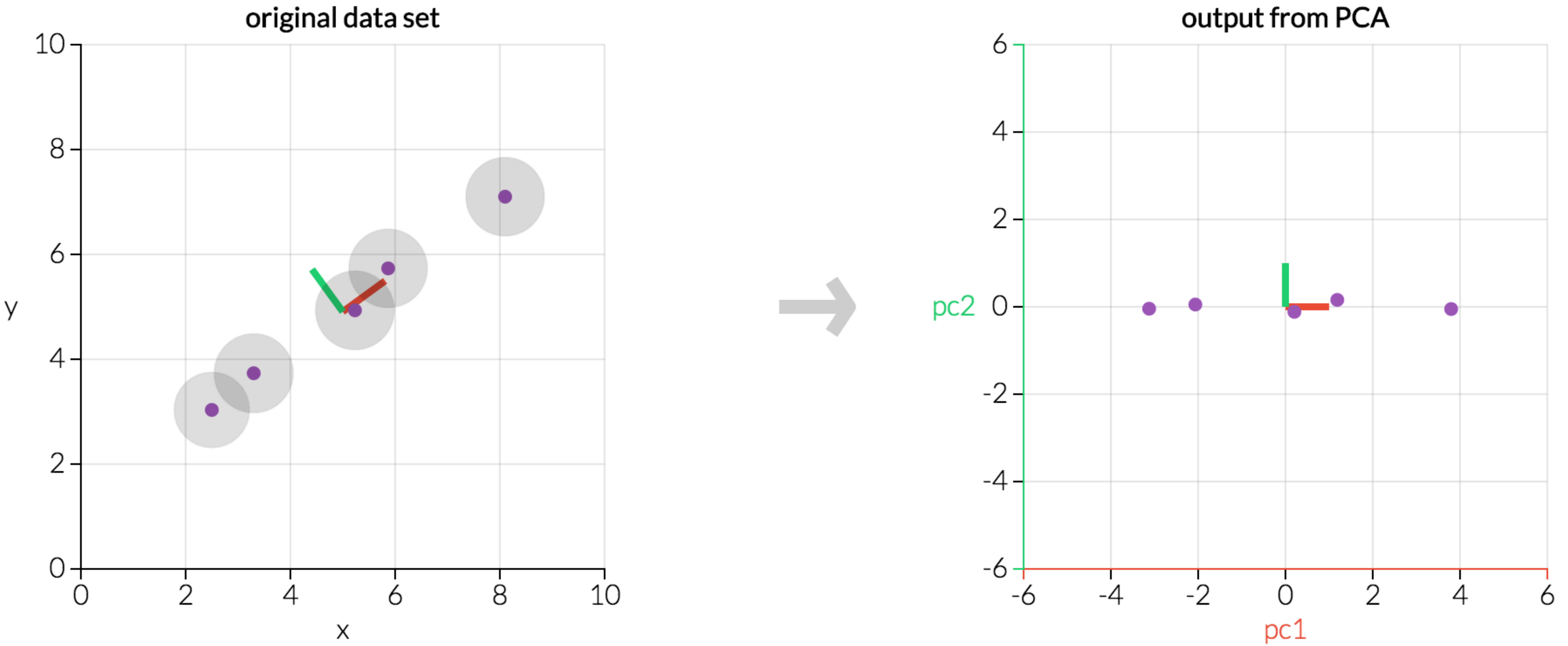
Plot lower dimensional space

Principal Component Analysis (PCA)

linear mapping, by order of variance



PCA


original data set
10
8
6
4
2
0
y
0
2
4
6
8
10
x
output from PCA
6
4
2
0
-2
-4
-6
pc2
-6
-4
-2
0
2
4
6
pc1

Multidimensional Scaling

Multiple approaches

Works based on projecting a similarity matrix

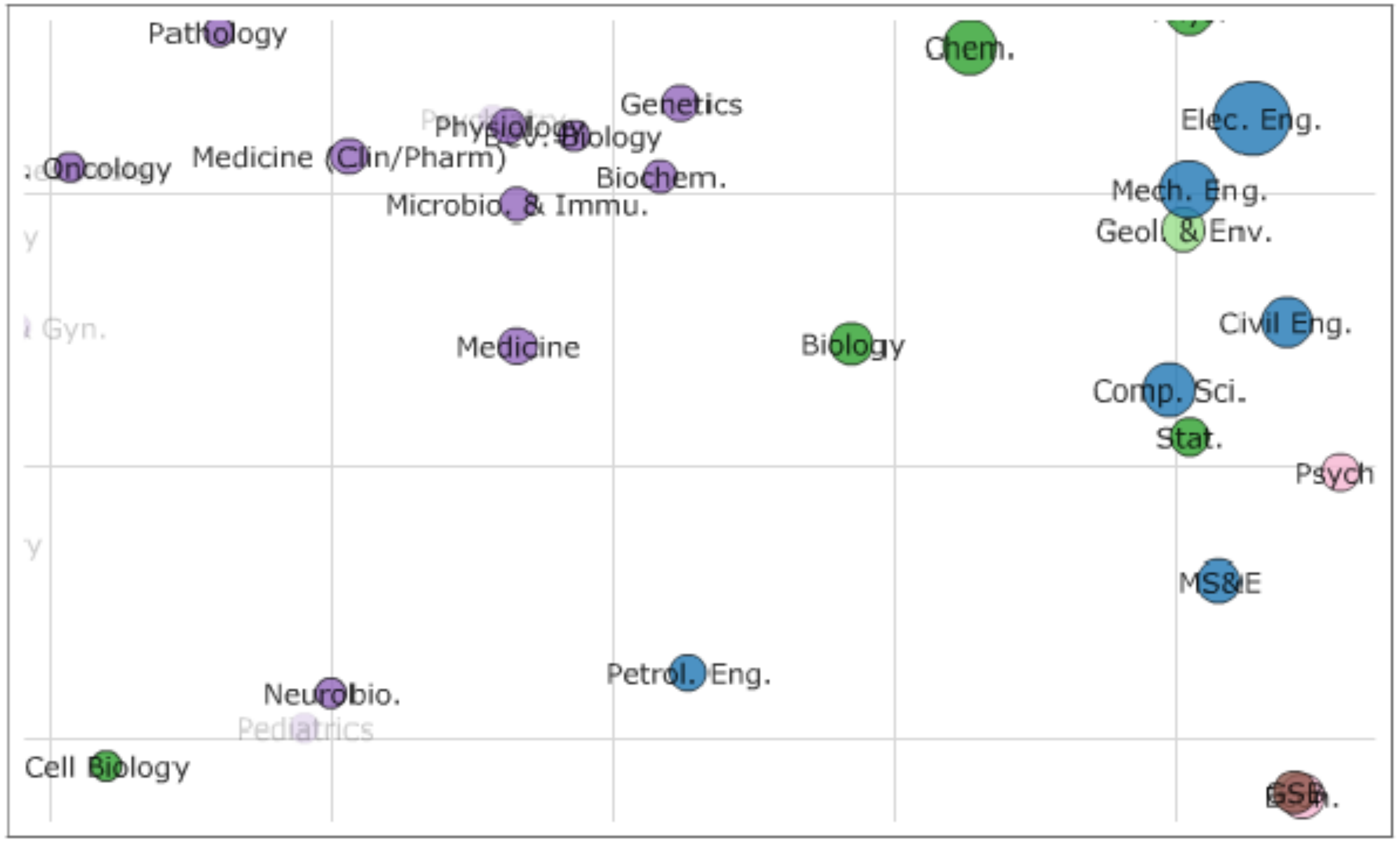
- How do you compute similarity?
- How do you project the points?

Popular for text analysis

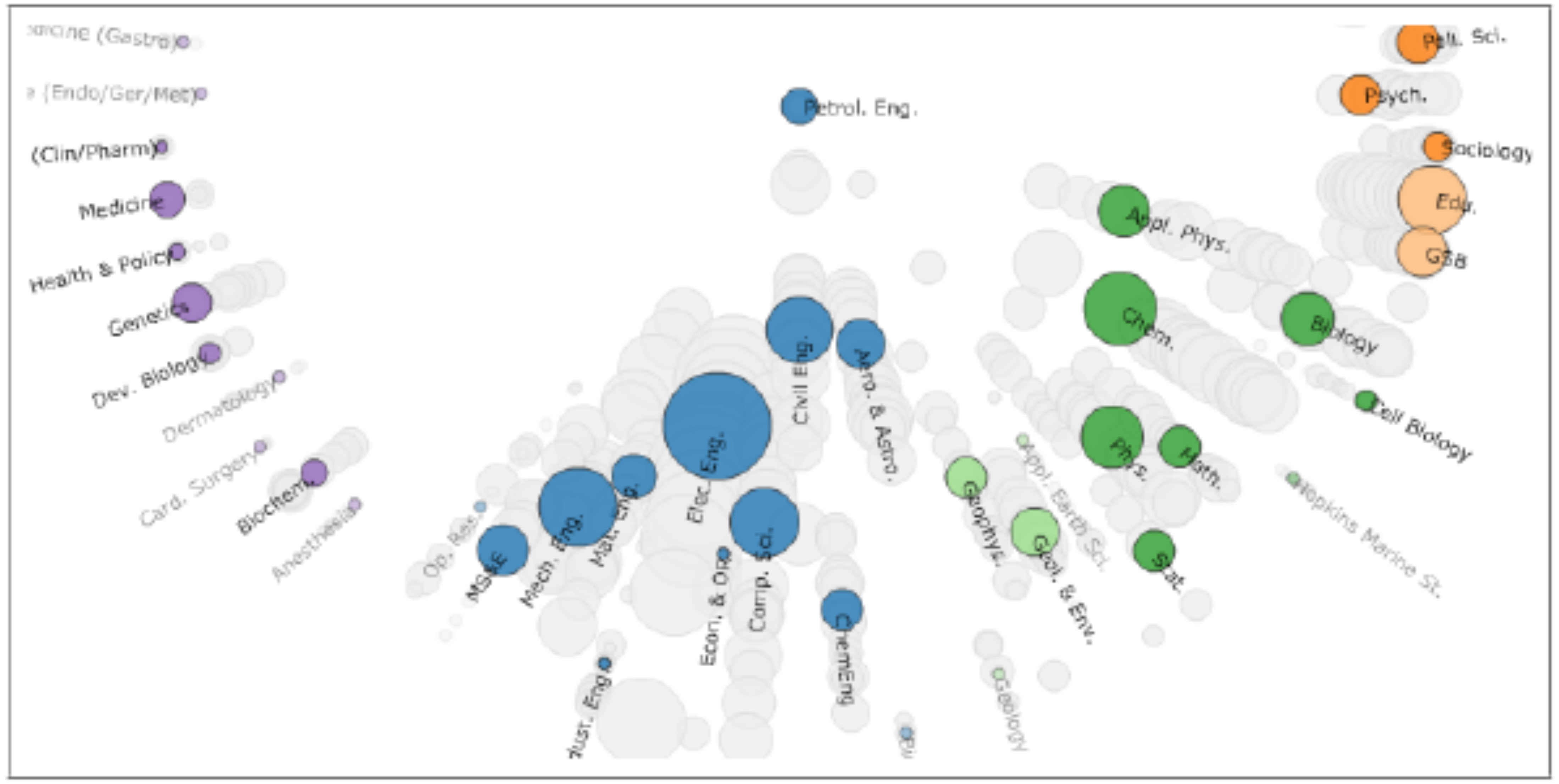
[Doerk 2011]

Can we Trust Dimensionality Reduction?

Topical distances between departments in a 2D projection



Topical distances between the selected Petroleum Engineering and the others.



[Chuang et al., 2012]

http://www-nlp.stanford.edu/projects/dissertations/browser.html

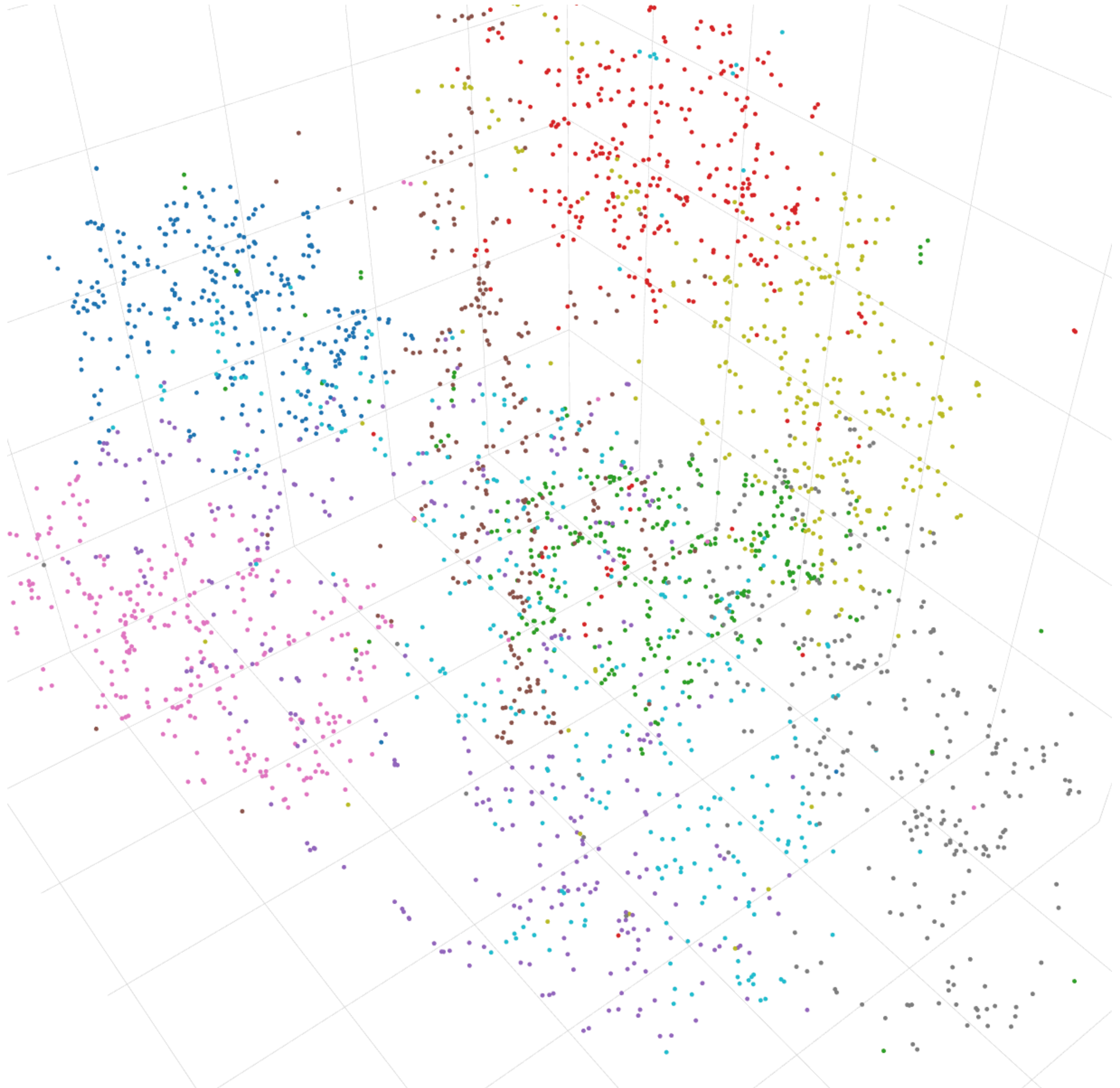
Probing Projections

http://julianstahnke.com/probing-projections/

t-SNE

t-distributed stochastic neighbor embedding

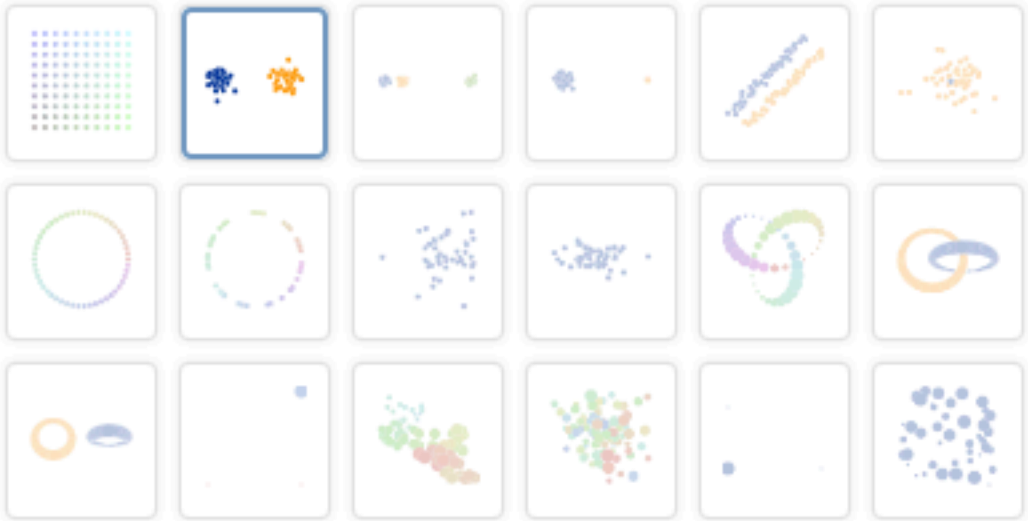
non-linear algorithm: different transformations for different regions



Visualizing data using t-SNE, Maaten and Hinton, 2008

How to Use t-SNE Effectively

Although extremely useful for visualizing high-dimensional data, t-SNE plots can sometimes be mysterious or misleading. By exploring how it behaves in simple cases, we can learn to use it more effectively.



Step 5,000

Two clusters with equal numbers of points.

Share this view

Points Per Cluster 50

Dimensions 2

Perplexity 10

Epsilon 5

MARTIN WATTENBERG
Google Brain

FERNANDA VIÉGAS
Google Brain

IAN JOHNSON
Google Cloud

Oct. 13
2016

Citation:
Wattenberg, et al., 2016

Understanding UMAP

Andy Coenen, Adam Pearce | Google PAIR

Dimensionality reduction is a powerful tool for machine learning practitioners to visualize and understand large, high dimensional datasets. One of the most widely used techniques for visualization is t-SNE, but its performance suffers with large datasets and using it correctly can be challenging.

UMAP is a new technique by McInnes et al. that offers a number of advantages over t-SNE, most notably increased speed and better preservation of the data's global structure. In this article, we'll take a look at the theory behind UMAP in order to better understand how the algorithm works, how to use it effectively, and how its performance compares with t-SNE.

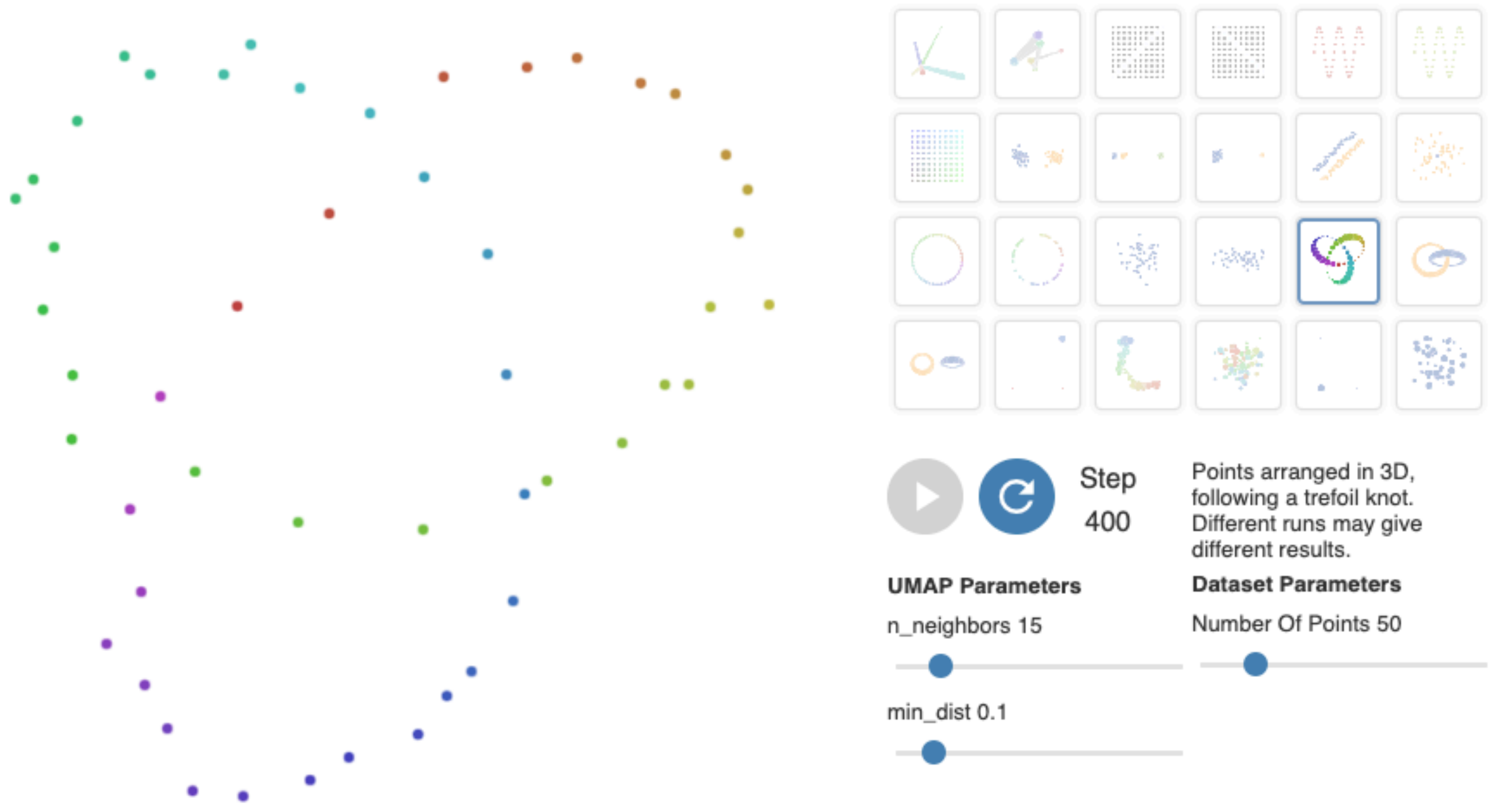
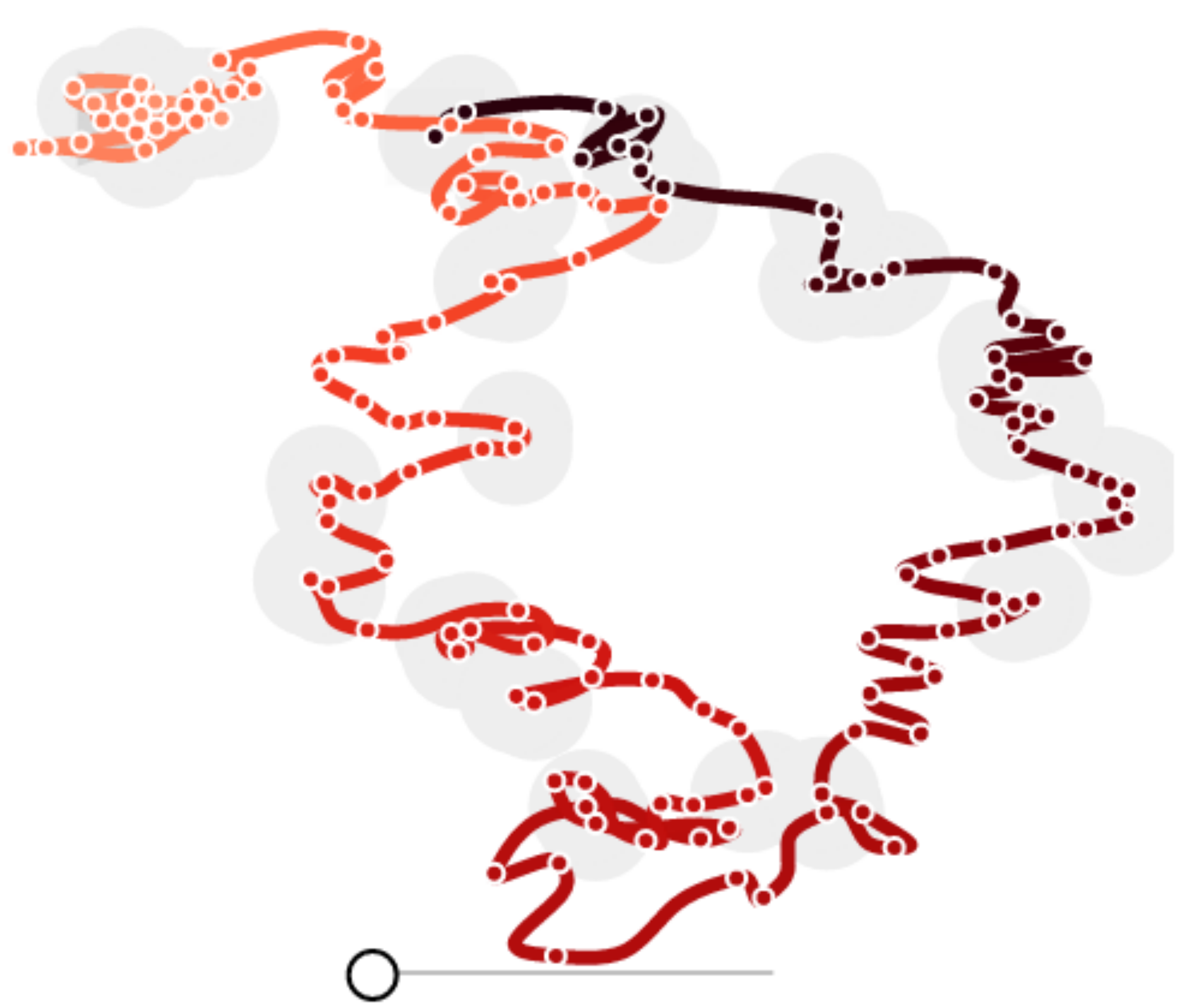


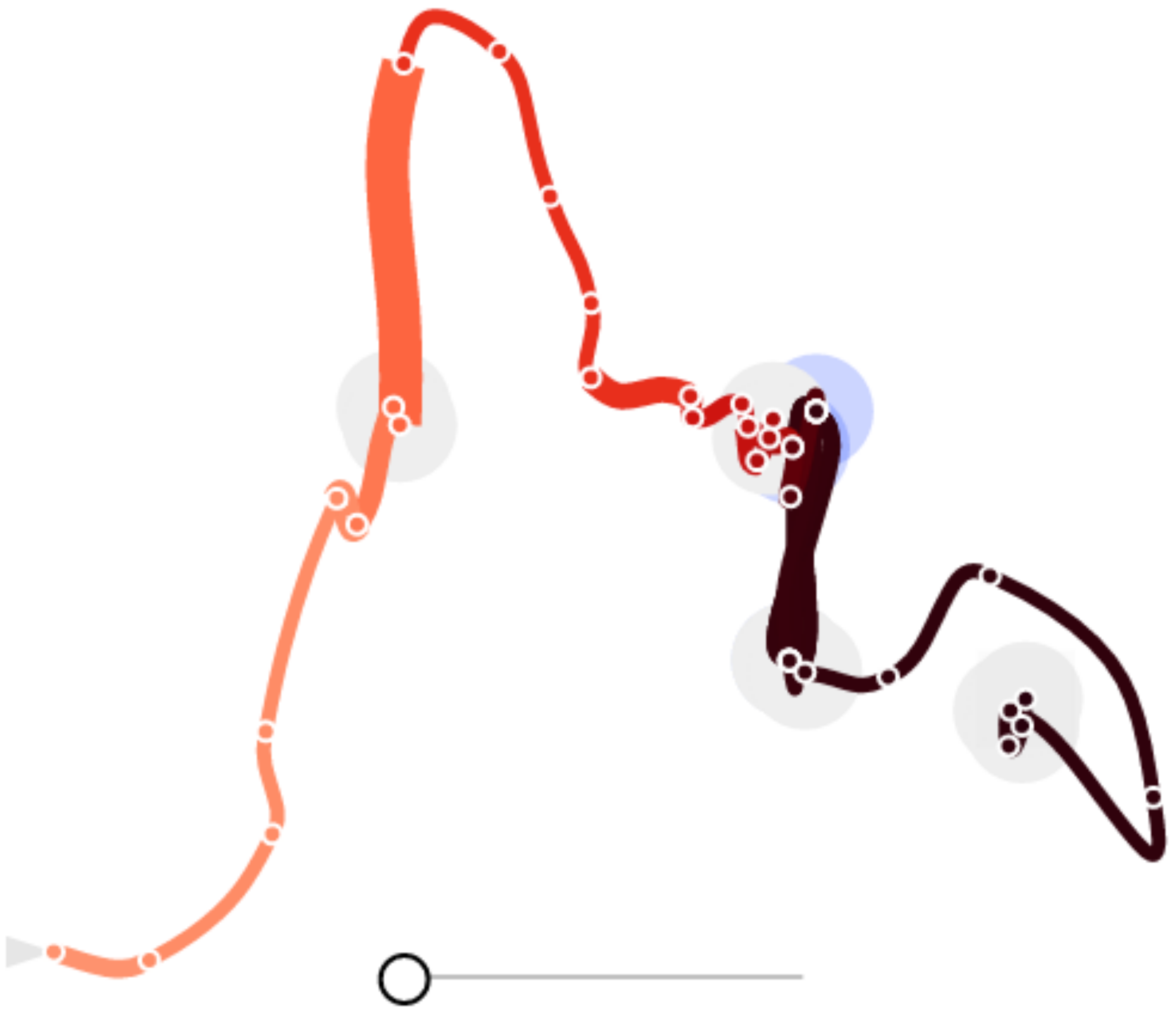
Figure 1: *Apply UMAP projection to various toy datasets, powered by umap-js.*

So what does UMAP bring to the table? Most importantly, UMAP is fast, scaling well in terms of

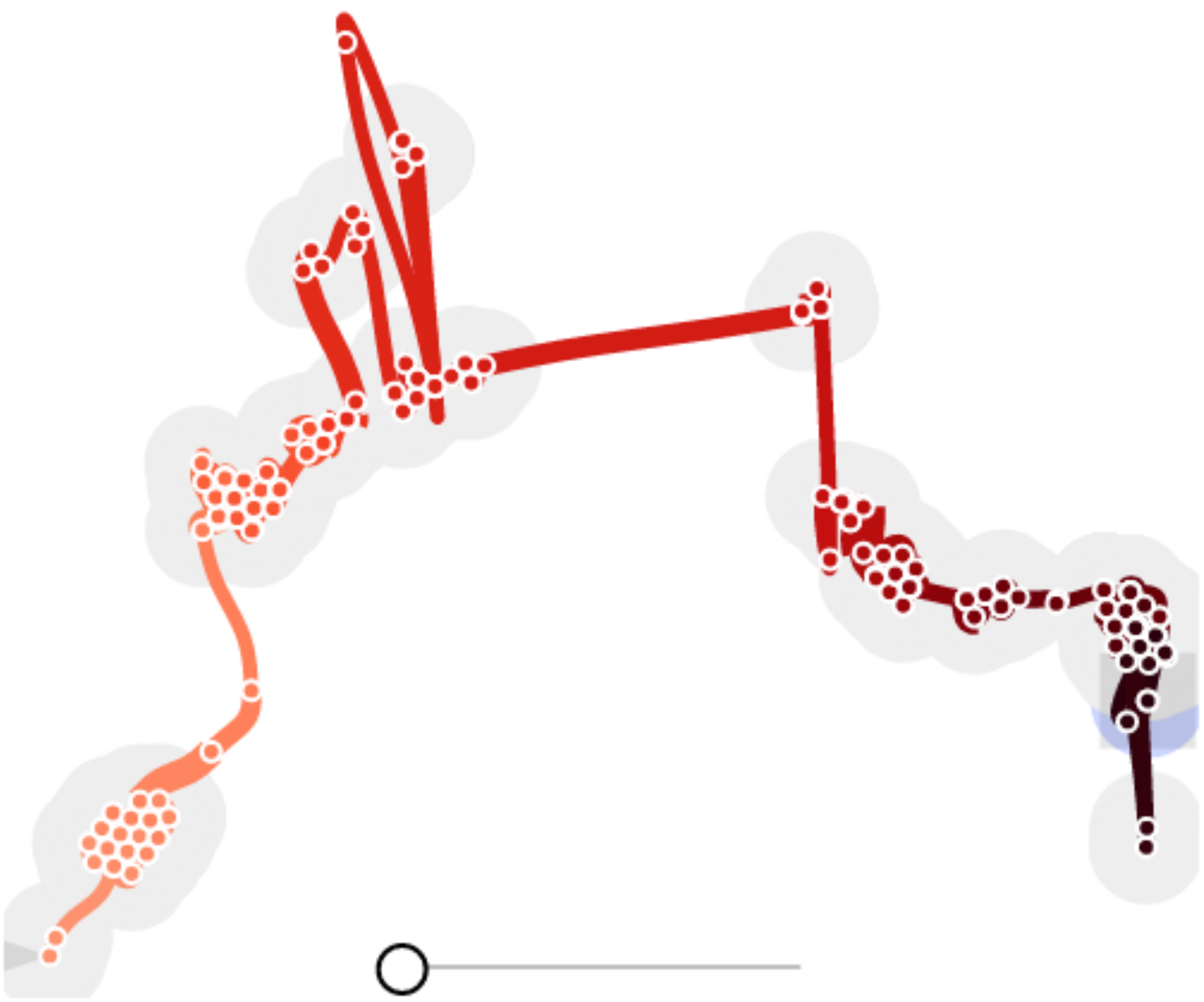
MDS for Temporal Data: TimeCurves



Video: Global Cloud Circulation (146)



Wikipedia: Chocolate (46)



Wikipedia: Palestine 200 1 (200)

http://aviz.fr/~bbach/timecurves/