Data Science & Drought

or How a Mathematician can be Useful in the Real World

Jordan Watts

Central Michigan University

November 13, 2018

Goal

From January to July 2018, I worked at the National Drought Mitigation Center, part of the University of Nebraska Lincoln, as a short-term postdoc.

One of the goals of my time there was to cluster drought stations in the continental United States geographically using many variables:

- monthly precipitation values,
- monthly temperature values,
- latitude and longitude,
- elevation,
- other derived features, such as drought indices, and comparisons with various oceanic and atmospheric indices.

Disclaimer!

I am a trained **pure** mathematician. I had very little statistical background; my area of research up until that point had been mainly in geometry and topology.

Besides this, I can program in a few languages, in particular Python and SQL.

I also taught myself the basics of topological data analysis beforehand, which is essentially simplicial algebraic topology.

Clustering Drought Stations

A **drought station** is a (regulated) location that produces accurate precipitation and temperature data.

Each station has coordinates given by latitude, longitude, and elevation.

The continental US has roughly 4000 such stations.

By clustering stations with similar data together,

1. more statistically-sound patterns can emerge,
2. and locations between drought stations can use more regional information to predict precipitation.

For example, perhaps a farmer wants to get an idea of just how dry the upcoming year might be, based on the region's precipitation history.

Clustering Drought Stations

The National Drought Mitigation Center provides data analysis of this type on their website:

Drought Risk Atlas: `http://droughtatlas.unl.edu`

Theorem: Singular Value Decomposition
Let A be a (real) $m \times n$ matrix. Then A decomposes as

$$A = PSQ$$

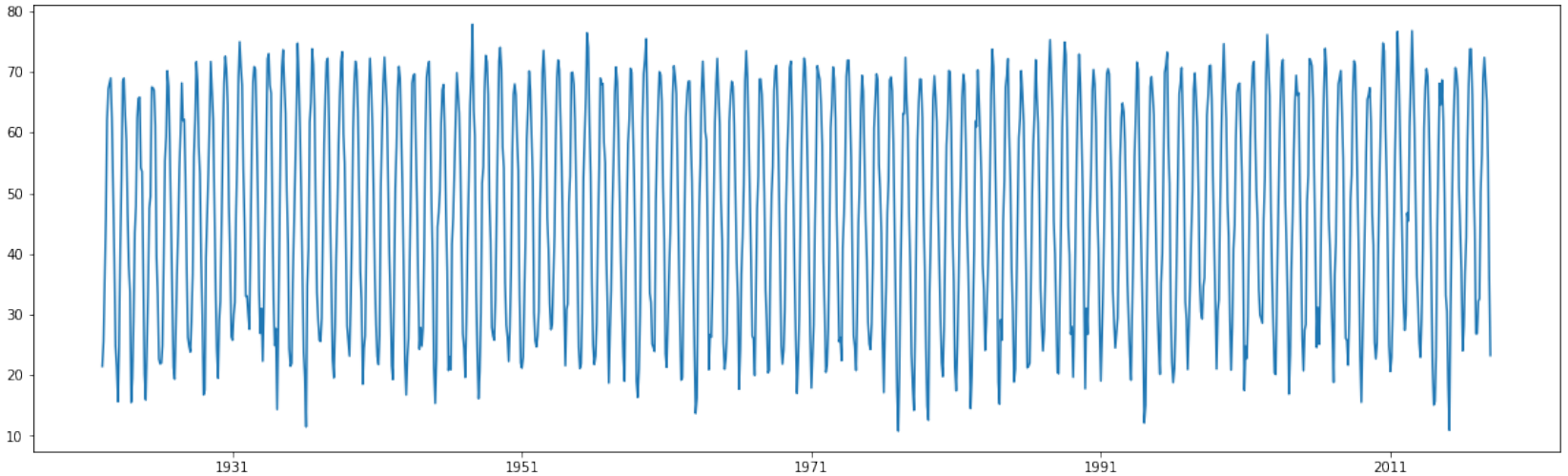
where P is an $m \times m$ orthogonal matrix, Q is an $n \times n$ orthogonal matrix, and S is an $m \times n$ diagonal matrix whose entries are non-negative and decreasing down the diagonal (the **singular values** of A).

That is, up to orthogonal changes of coordinates in both the domain and codomain, every matrix is diagonal with non-negative entries.

Note: The rank of A equals the rank of S.

Example: Temperature Data

Consider the average monthly temperature in Mount Pleasant, Michigan, recorded January 1922 until December 2017.



One can clearly see the annual cycle in temperature in this data.

Example: Temperature Data

For each year, form a vector in $\mathbb{R}^{12}$ whose components are the (ordered) temperatures of each month, January-December. Form a matrix whose columns are made up of these vectors (this would be $2017 - 1922 + 1 = 96$ columns).

$$\begin{bmatrix} 21.468 & 21.355 & 15.518 & 15.951 & 21.887 & 19.366 & \\ 25.553 & 15.625 & 19.5 & 24.857 & 22. & 27.785 & \\ 35.161 & 24.592 & 28.952 & 33.452 & 24.936 & 36.517 & \\ 45.816 & 41.334 & 43.5 & 47.533 & 39.117 & 44.117 & \\ 62.096 & 52.306 & 47.968 & 49.484 & 55.419 & 53.968 & \\ 67.2 & 68.466 & 62.4 & 67.516 & 59.367 & 60.383 & \\ 68.049 & 68.968 & 65.661 & 67.403 & 70.194 & 68.177 & \cdots \\ 69. & 62.952 & 65.855 & 66.98 & 67.597 & 61.968 & \\ 62.3 & 58.733 & 54.217 & 61.216 & 58.133 & 62.166 & \\ 51.065 & 46.822 & 53.565 & 39.645 & 46.291 & 52.806 & \\ 39.35 & 38.1 & 37.517 & 33.233 & 34.45 & 37.684 & \\ 24.919 & 33.726 & 20.241 & 22.694 & 21.306 & 26.242 & \end{bmatrix}$$

Example: Temperature Data

In a world where the temperatures were exactly the same every year, what would the rank of this matrix be?

In reality, we cannot expect a rank-1 matrix; typically the rank will be 12 since there will most likely be 12 linearly independent columns in the matrix.

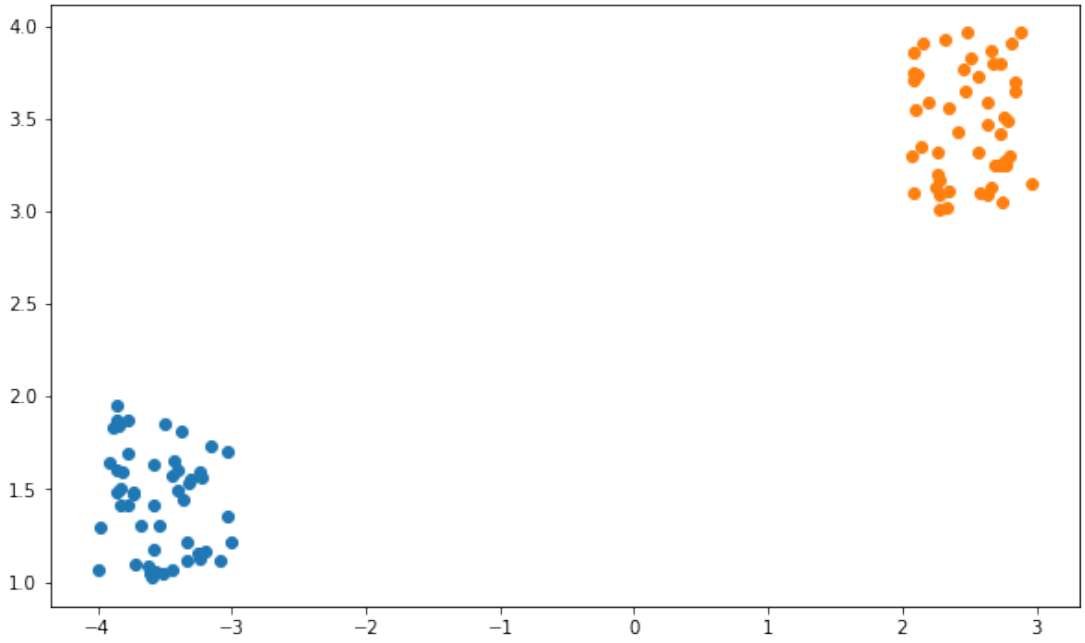
However, consider the singular values of the matrix:

$$\begin{array}{cccc} 1698.52237955 & 50.22183146 & 47.91601992 & 40.89820659 \\ 34.61589362 & 33.35811886 & 29.25988025 & 28.02671064 \\ 23.8569622 & 21.51751487 & 20.48817112 & 16.81684667 \end{array}$$

While there are 12 non-zero values, one of them is very prominent (the second singular value is only 3% of the first).

Example: Clustering

Consider the following two clouds of data sitting in the plane.



Treating each point like a vector and forming a matrix with these vectors as columns will yield a matrix of rank 2; this is not surprising.

Example: Clustering

However, if we have two clouds in a higher dimensional space, say 5-dimensional, then forming a similar matrix, we will have a rank 5 matrix.

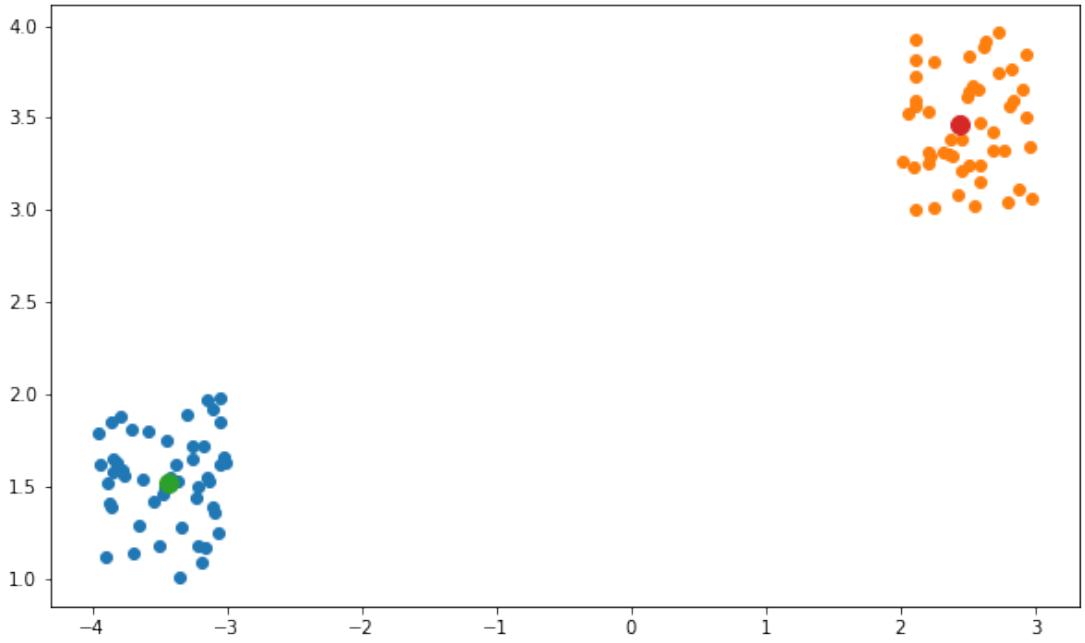
$$\begin{bmatrix} -3.49 & -3.83 & -3.12 & \cdots & 2.53 & 2.33 & 2.50 & \\ 1.43 & 1.07 & 1.90 & \cdots & 3.85 & 3.20 & 3.41 & \\ 17.13 & 17.44 & 17.67 & \cdots & -2.52 & -2.60 & -2.60 & \cdots \\ -16.60 & -16.09 & -16.54 & \cdots & -4.51 & -4.59 & -4.23 & \\ 7.52 & 7.31 & 7.23 & \cdots & 10.94 & 10.69 & 10.71 & \end{bmatrix}$$

Now consider the singular values.

$$183.35,\ 81.93,\ 2.85,\ 2.72,\ 2.66$$

Example: Clustering

By setting the three smaller singular values to 0, we can obtain the following points:



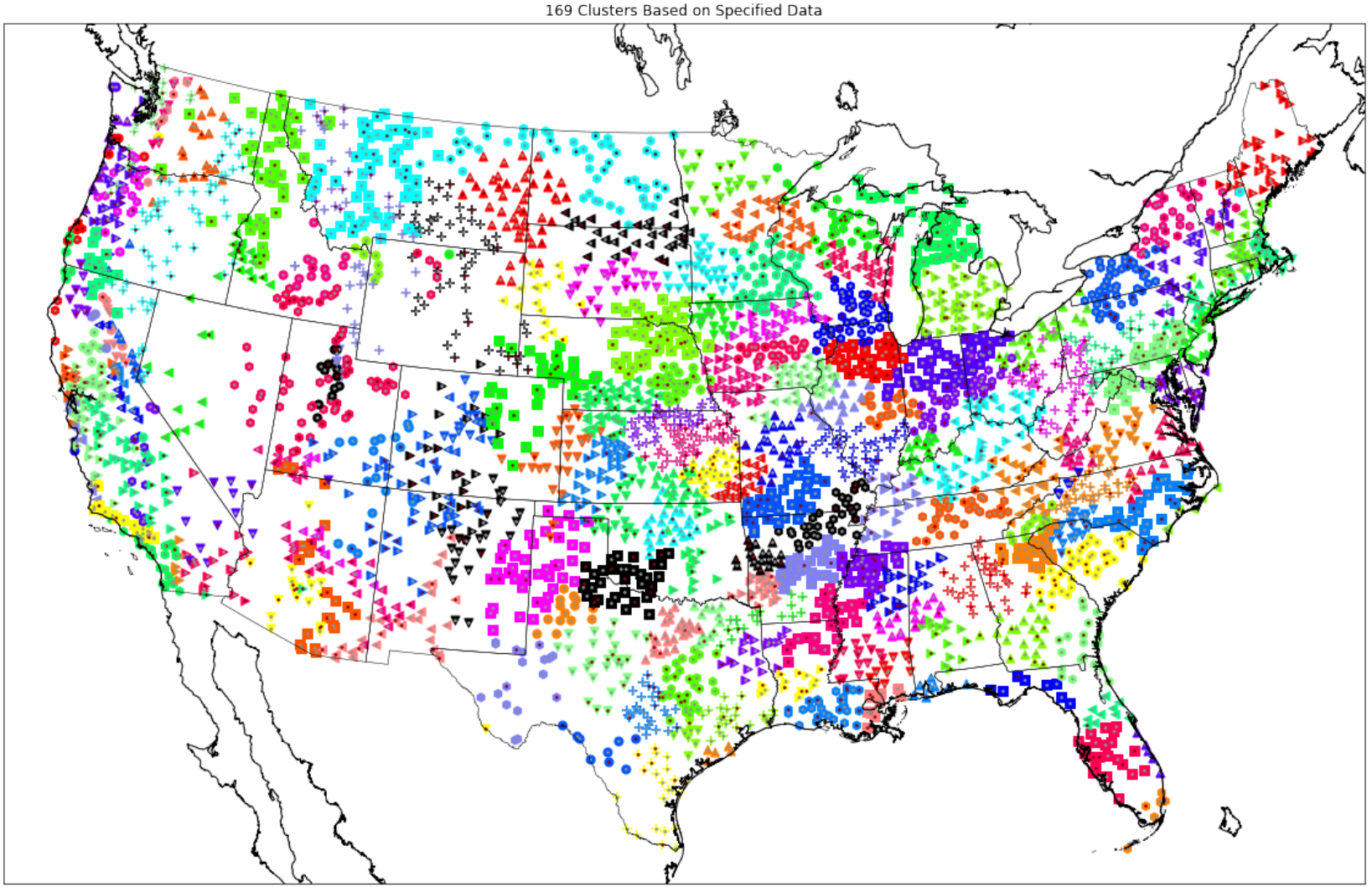
Clustering Drought Stations

By treating the latitude, longitude, perhaps elevation, monthly precipitation values, perhaps monthly temperature values, we can form thousands of vectors, one for each station, in a high dimensional space (one dimension each for latitude, longitude, and elevation, one each for each month of precipitation, one each for each month of temperature, etc.).

Putting these into a giant matrix and applying singular value decomposition, throw away smaller singular values until you have reached an error threshold (explained variance) that is acceptable.

The remaining singular values will correspond to clouds of data, which in turn correspond to regions of drought stations.

Clustering Drought Stations



Clustering Drought Stations

Issues:

1. Some of the clusters are too small (<5 stations).
2. Some of the clusters are disconnected – should they be?
3. Some of the stations have distributions that do not match the other stations (discordancy).

This used to be fixed by hand, with “expert subjectivity”.

My goal was to do most of this by machine.

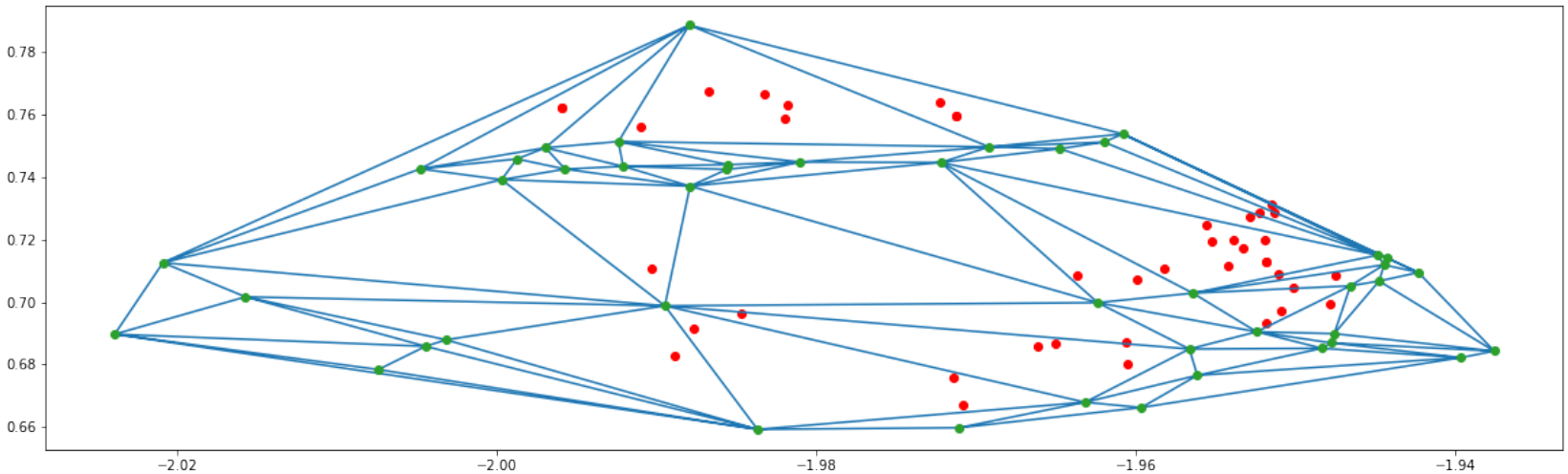
Cardinality, Connectivity, and Concordancy

Clusters of stations were sent through a “CCC Machine”, where the three Cs stand for cardinality, connectivity, and concordancy.

Clusters with too few stations were disbanded, and their constituent stations were merged into other clusters using a “nearest neighbour” algorithm.

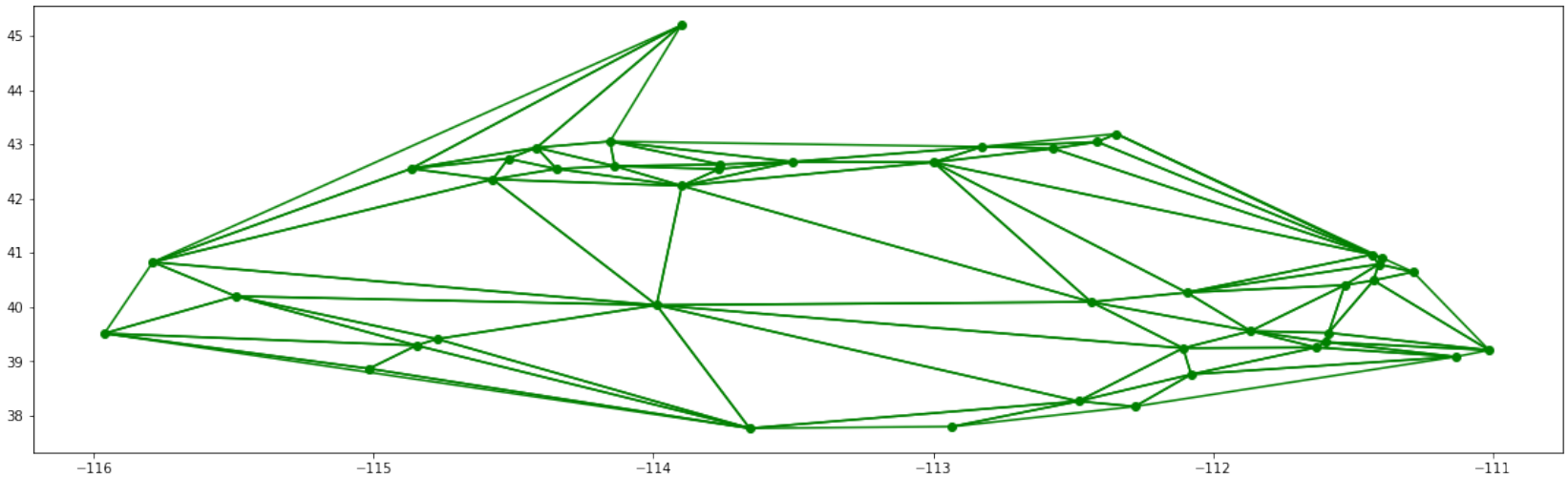
Cardinality, Connectivity, and Concordancy

To deal with connectivity, each cluster (green vertices) was triangulated (blue edges). Stations contained inside the cluster geographically, but not part of the cluster itself, are the red vertices.



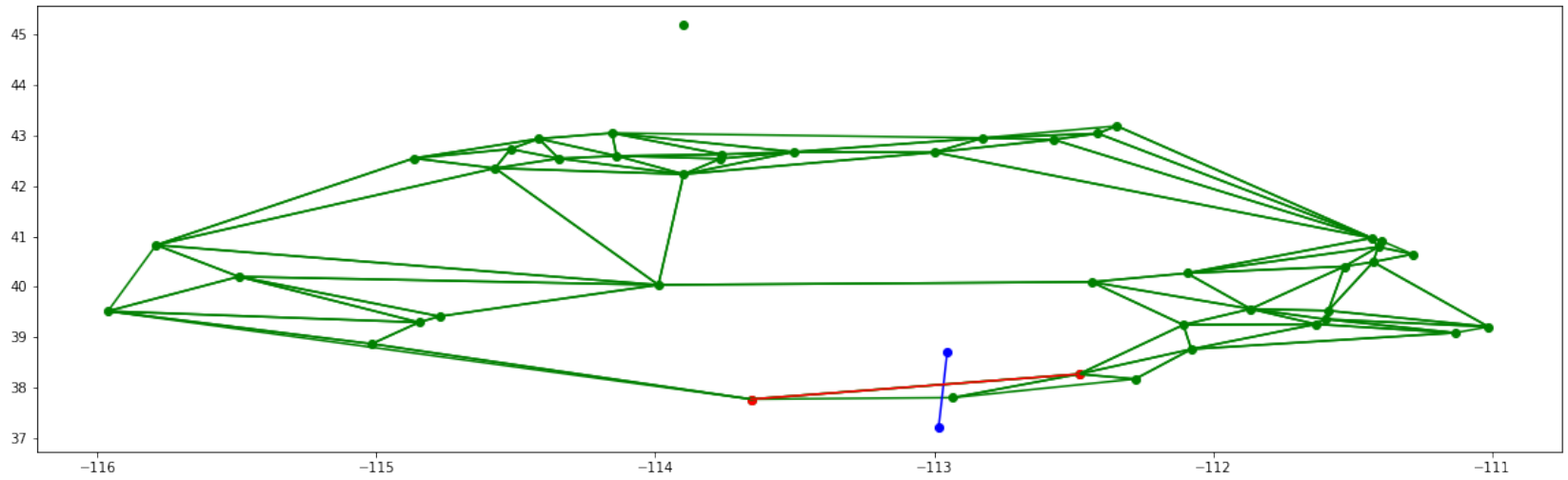
Cardinality, Connectivity, and Concordancy

1.) When there were red stations (stations outside the cluster) in a triangle adjacent to the boundary of the cluster, the corresponding boundary edge was removed.

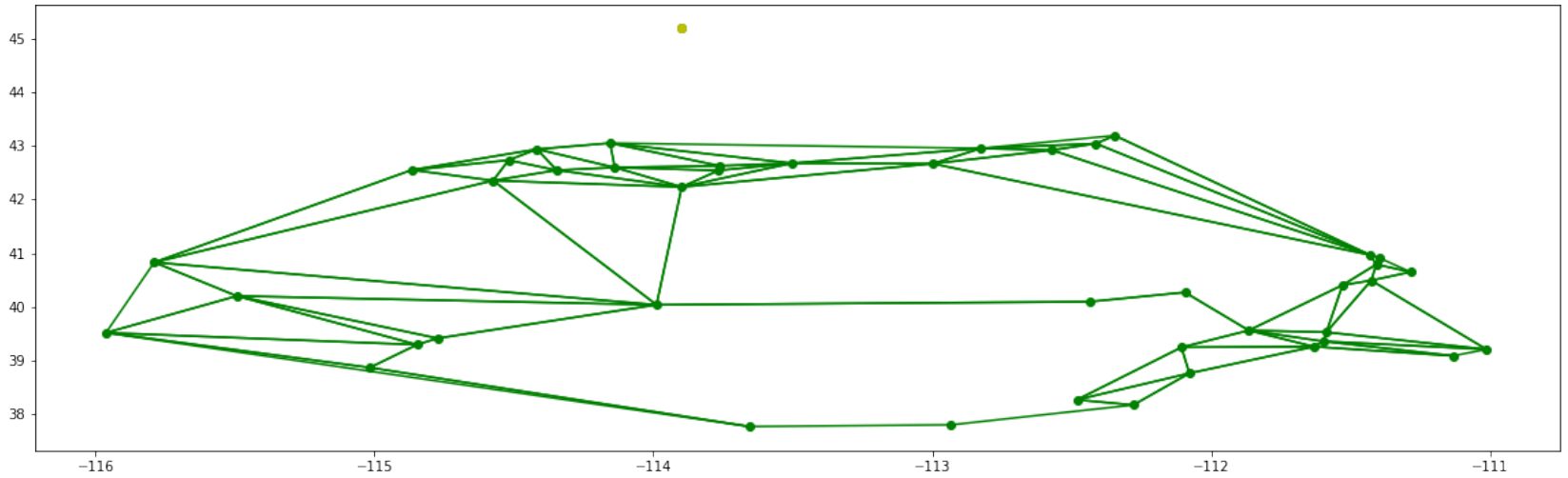


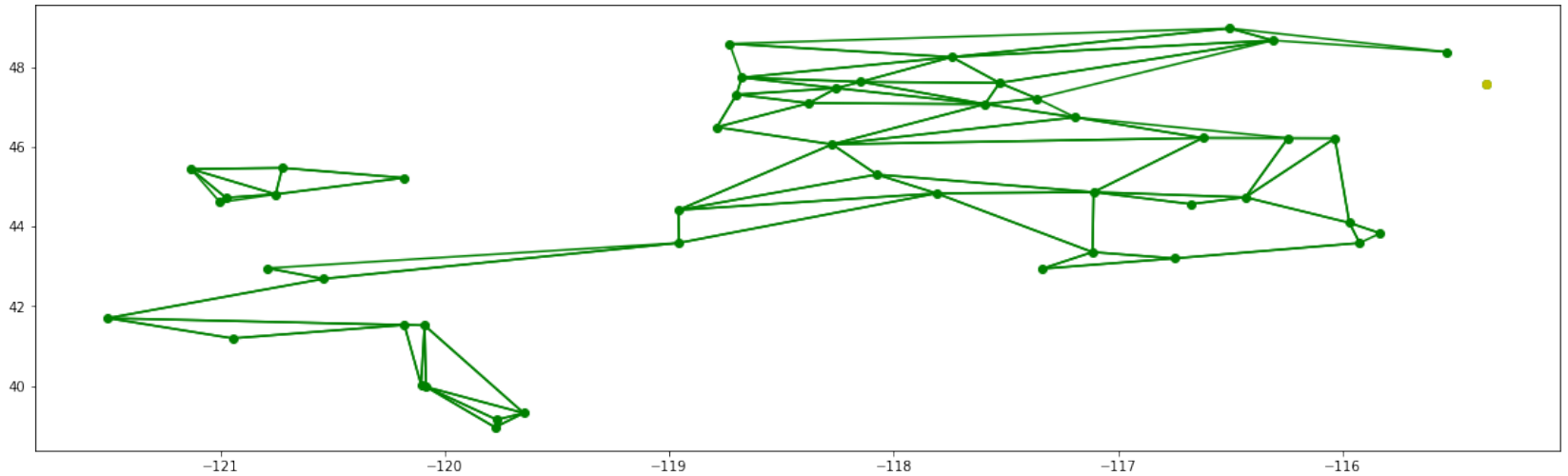
Cardinality, Connectivity, and Concordancy

2.) When two stations (blue) where close on either side of an edge (red), this edge was deleted.



Cardinality, Connectivity, and Concordancy



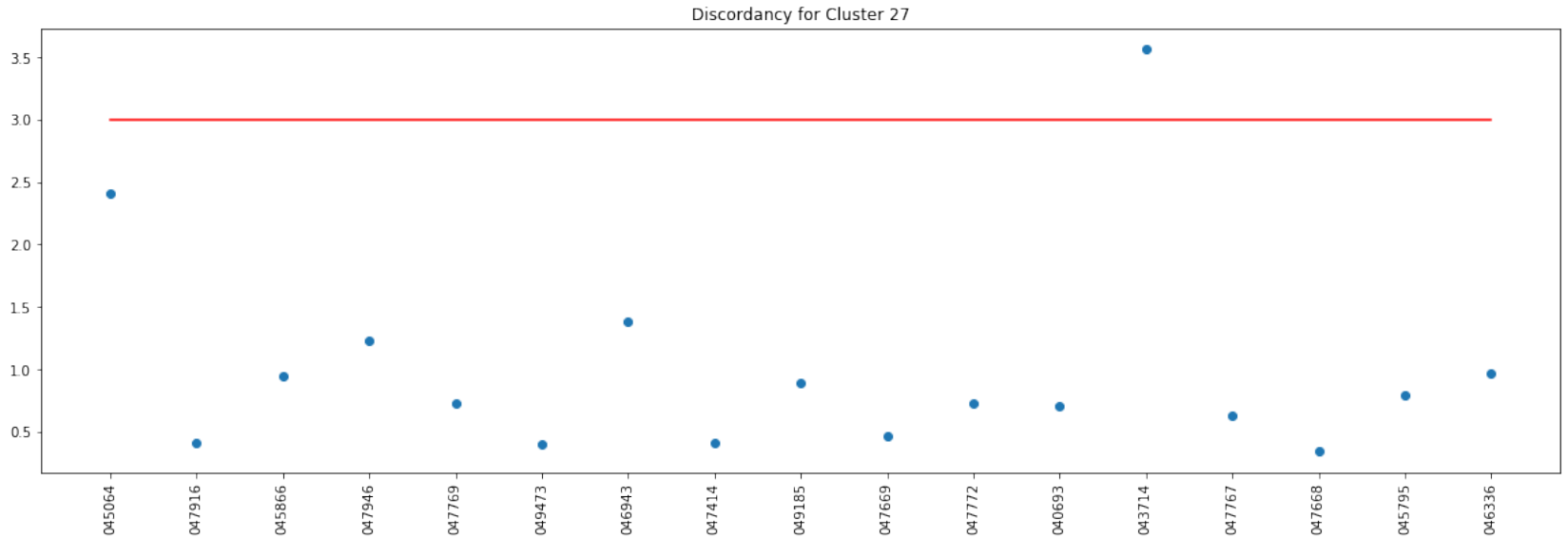


Cardinality, Connectivity, and Concordancy

Sometimes new clusters were created using this method, and sometimes some stations were broken off and absorbed into other nearby clusters.

Cardinality, Connectivity, and Concordancy

Finally, a measure of discordancy was used to determine which stations in a cluster (if any) had data distributions that were too different from others in the cluster.



These stations were removed and a nearest neighbour algorithm was used to send them to other clusters (possibly the same original cluster).

The program cycled through each of these subroutines many times until the clusters stabilised. (To ensure stabilisation, the program kept track of how many times a station moved in and out of a cluster, and prohibited stations from moving in this way more than k times.)

Homogeneity

After the CCC Machine was done, each cluster was tested for **homogeneity**: does the data from all of the stations in a cluster fit into a reasonable distribution?

All heterogenerous clusters were broken up, and all the corresponding stations were sent through the clustering process, as well as the CCC machine.

This process continued until the vast majority of stations belonged to a homogeneous cluster.

Probabilistic Perspective

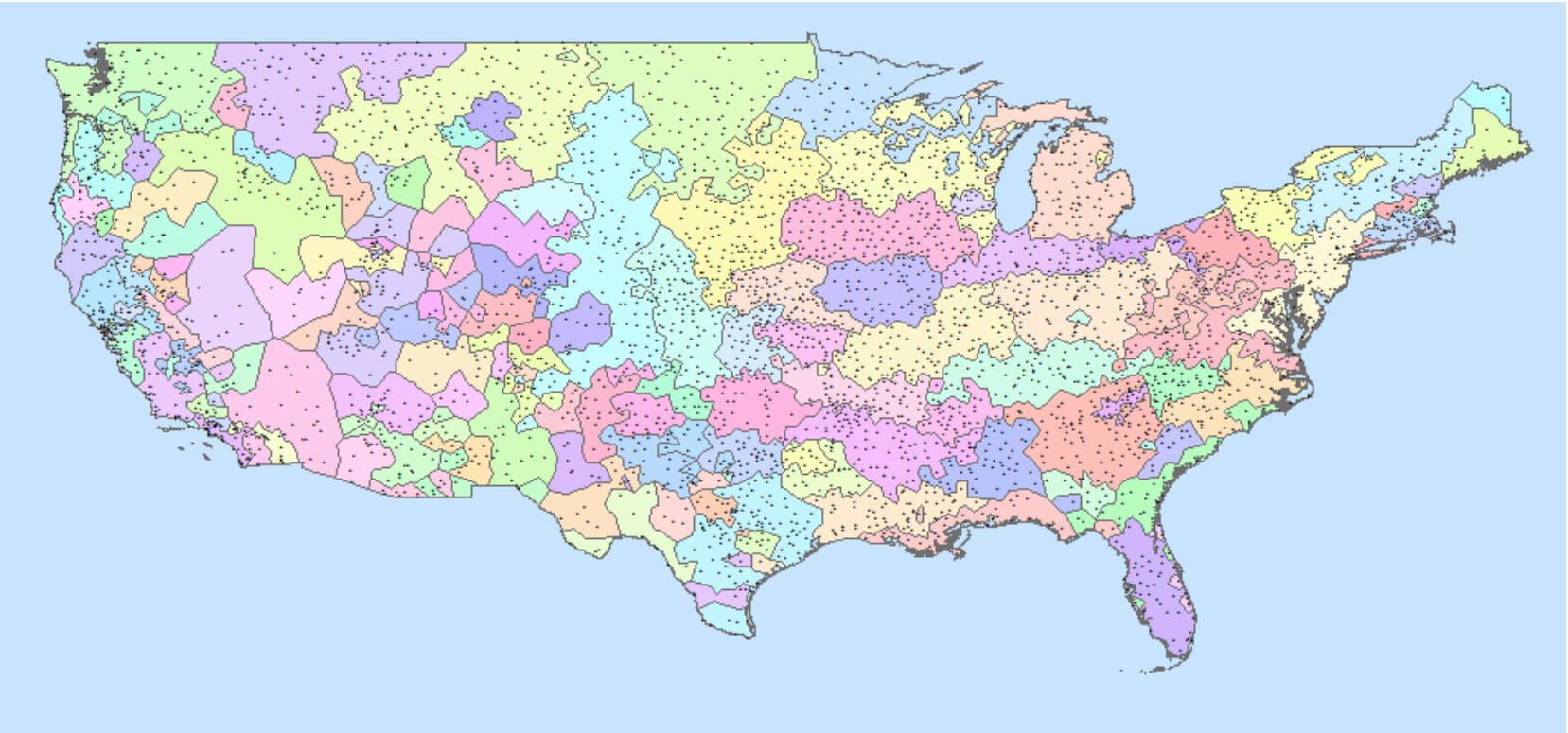
In turns out that the clustering procedure (in particular, the CCC Machine) can produce different clusterings of the stations; the main issue being areas of the country with sparse station coverage.

This means that given any two stations, there is a probability that the they would end up being in the same cluster.

By placing two stations in the same final cluster if they were put together, say, 80% of the time by the program above (and using a nearest neighbour algorithms for clustering leftover stations), one obtained reasonable clusters that have at least 5 stations each, were typically connected, and typically concordant and homogeneous.

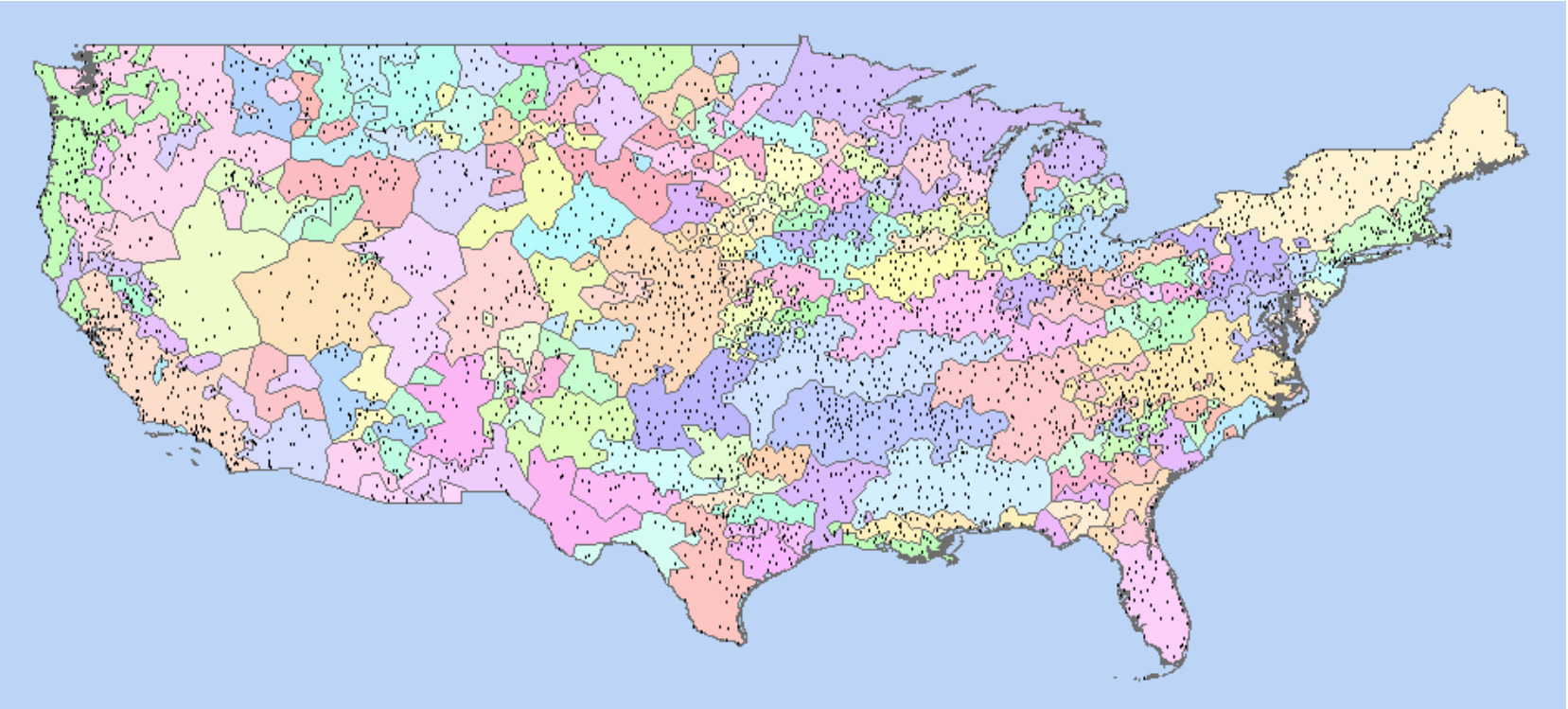
Maps

Clustering based on summer precipitation, temperature, elevation, latitude, and longitude.



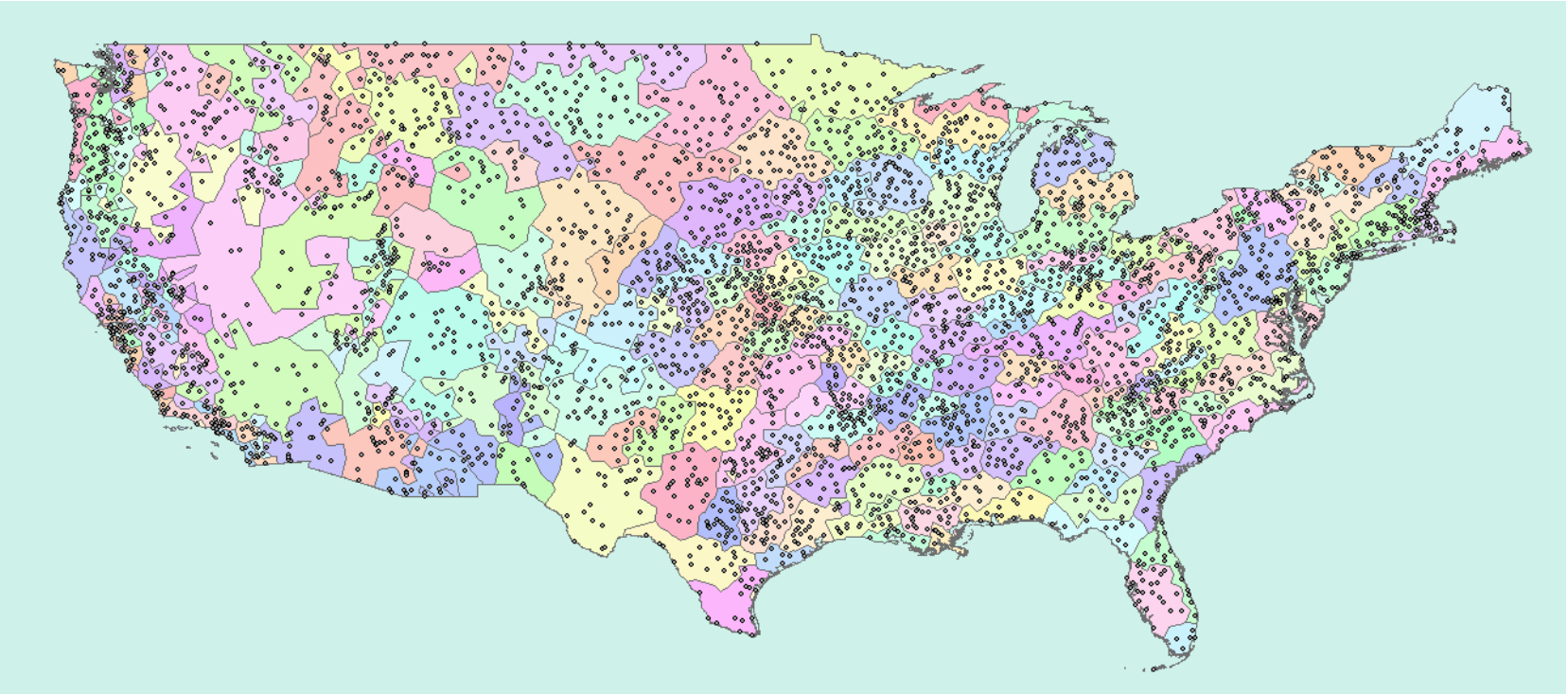
Maps

Clustering based on spring precipitation, temperature, elevation, latitude, and longitude.



Maps

Clustering based on monthly precipitation and temperature for the entire year, along with elevation, latitude, longitude, and other features considered by experts.



Thank you!