Ryan Kaplan

Data Analytics Semester Project

1) Cluster analysis is a technique for analyzing data through examining the groups, or "clusters", that tend to form within large data sets. If these groups were given it would be a simple classification task, however this technique is unsupervised, with no such labeled data points.(1) Instead, we group data points together in terms of similarity. I was interested in continuing my work on the California environmental research paper, which involved applying cluster analysis to several relevant datasets, by comparing two of the most prominent clustering techniques; k-means and hierarchical clustering. The idea I want to investigate would thus be, how do the two methods differ in how they cluster this data? Specifically, how do they differ both in terms of a hard cluster evaluation metric, and how do they differ in the maps they can be used to make. The three datasets used are the California precipitation, soil moisture, and solar radiation datasets, all obtained from NASA, and all three datasets contain coordinate information as one of each data point's features, so once we've assigned a cluster label to each point, we can examine them overlaid on a map of California, and see how different regions are classified. I do not expect to find tremendous differences in the maps made by the two methods. For example, I would be surprised to see a large section of desert classified as a high precipitation area regardless of the clustering method, however I do look forward to seeing the differences where they exist.

2) As mentioned previously, the three datasets used were the California precipitation, soil moisture, and solar radiation datasets. The California precipitation dataset consists of data points measured from between June 2000 and September 2021, and each point consists of the date,

latitude and longitude, and the average monthly precipitation in that area. The California soil moisture dataset has points from the same time range, and each point has the date, latitude/longitude, soil moisture density in kg/m^2, soil moisture availability percentage, as well as average surface temperature. Interestingly, the solar radiation dataset ends at the same time, September 2021, but has data points going back to January 1980. Here, each point consists of date, latitude/longitude, and surface absorbed longwave radiation (LWGAB). I should note, all three datasets describe the same California region, shown here:

Figure 1: Blank region map for the California data

All three datasets were given to me as part of my work on the California environmental research paper I've been assisting with, and since they come from NASA we can have high confidence in its veracity. I think environmental data is a natural fit for clustering work, because it fits the way we naturally think about environments. We think of wet regions, dry regions, hot regions, etc, and since grouping areas into biomes comes so naturally, it's interesting to see how a machine learning algorithm approaches the same task. I know that compared to me, someone from India would have wildly different definitions of hot and cold weather, or a very different reckoning of

what a lot of rain is from someone living in Egypt, so using environmental datasets to look at the difference between two clustering algorithms will hopefully prove enlightening. Additionally, this affinity for clustering is not just in the eye of a human beholder, since similar environmental conditions do inherently tend to appear together. That is to say, adjacent regions tend to have similar conditions, and if conditions are such that a certain region is very dry, it is most likely that an adjacent region will be very dry as well. In other words, we should expect before even examining the data that like regions will be grouped together, and thus make for good clustering material.

3) As mentioned previously, since the data comes from NASA, we can place a high amount of confidence in it. The only example I saw where one would consider cleaning the data came from the solar radiation dataset, where a few of the data points lacked a date. I chose to include them, as I did not include the date in my clustering, and they were otherwise normal data points, and thus valuable to include in clustering. I did check that the coordinate information was indeed where I thought it was, and while it does contain more of Nevada and the ocean than I originally thought, the coordinates did line up. Note that because it is necessary for clustering, these values are all normalized. For the precipitation dataset, the main relevant feature is of course average monthly precipitation:

Minimum	-0.603
1st Quartile	-0.531
Middle Quartile	-0.346
3rd Quartile	0.075
Maximum	14.787

While we see a large gap between our third quartile and the maximum value, there are actually many high precipitation values represented. Since the data points take place over a large period of time, there are bound to be periods of unusually high precipitation, especially compared to the usual very low values. As such, these high values should not be considered outliers, but legitimate realities of the California climate, where sometimes it does rain heavily. Other than that, the variable range seems reasonable. Next we move on to the soil moisture dataset, where we have three variables with very similar distributions. First, we have raw soil moisture, as density:

Minimum	-1.728
1st Quartile	-1.72
Middle Quartile	0.570
3rd Quartile	0.585
Maximum	0.678

We have a reasonable distribution all around, with some values skewed towards the bottom. These much lower values represent the ocean, which was considered to have a soil moisture of negative 9999… Since this will make up a solid portion of our map, and cannot be considered an inaccurate reading, (there is no soil after all) we must keep these values. The next variable to examine for the soil moisture dataset is the soil moisture availability percentage.

Minimum	-1.728
1st Quartile	-1.72
Middle Quartile	0.577
3rd Quartile	0.579
Maximum	0.589

We see a distribution that is nearly identical to our soil moisture density, which is to be expected. Again, the ocean throws it off somewhat, but we must keep these data points in for the same reasons as before. Finally for soil moisture, we have the soil temperature:

Minimum	-1.728
1st Quartile	-1.72
Middle Quartile	0.574
3rd Quartile	0.576
Maximum	0.585

We should not be surprised to see another nigh-identical distribution, as the temperature of soil is going to be one of the most important factors that determines soil moisture. It's difficult to hold moisture after it evaporates, after all. Out last dataset, solar radiation, has just one variable other than latitude and longitude for us to examine:

Minimum	-2.89
1st Quartile	-0.72
Middle Quartile	0.148
3rd Quartile	0.680
Maximum	3.731

We see a very balanced distribution, with perhaps slight trends to both extremes. I suspect this is due to the long period of time during which the data was collected. Solar radiation values surely

differ between the summer and winter, which will naturally inflate the further values. Other than that, the minimum is about as far from the median as the maximum value is, and the distribution looks good.

4) The first clustering method, k-means, was implemented using SSE. SSE, or Sum Squared Error, looks to minimize the total squared distance, or error, between each node and its cluster's centroid, and we want the number of clusters which produces the minimum SSE. Using k-means to actually generate *n* clusters, we initialize *n* randomly placed centroids, and assign every data point to the closest one, and determine our SSE. Next, we set each centroid to the average value of the data points assigned to it, and repeat the process until our centroids stop changing, and take the *n* value with the lowest SSE. We can check our work using a similar process but with Silhouette Score, which instead examines the average distance between same-cluster data points and the average distance between it and the points in other clusters.(2) In all cases, we got the same optimal number of clusters using SSE and Silhouette Score. Hierarchical clustering is also an iterative process, however unlike k-means, while it is going on, the number of clusters is variable.(3) Initially, every single data point is its own cluster, and at each step, the two most closest clusters are merged. As with k-means, we determine the closest cluster using SSE between the elements of our cluster, and every other. Of course, if we keep going in this manner we eventually end up with a single cluster, however when the distance between clusters is large enough, we stop merging. We determine this distance the same way as before, and so end up with the same number of clusters for each dataset. Of course, as mentioned previously, data points must be normalized before this process can take place, since some attributes may have a far greater range than others. If you had one feature between zero and a thousand, and another

that's either zero or one, naively clustering without normalization could end up just grouping the points into 2 clusters based solely on the second feature. Finally, I evaluated the models using silhouette score, and two different metrics of distance with which to calculate it; euclidean and manhattan. Note that a higher silhouette score is better, as it represents farther apart and well-defined clusters. Possible scores range from -1 to +1. Euclidean distance can be thought of as “normal” distance, like if you laid a ruler directly from one point to another, whereas Manhattan distance can be thought of like city blocks on a grid, where diagonal movement is twice as far as horizontal movement. For the precipitation dataset, k-means gives us two, well-defined clusters:

Centroid	Avg. Precipitation	# Power Plants
A	0.3182	460
B	-0.2751	320

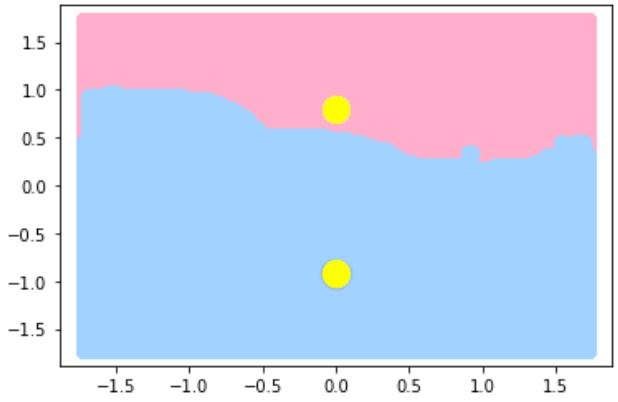

Figure 2: Precipitation centroids (k means)

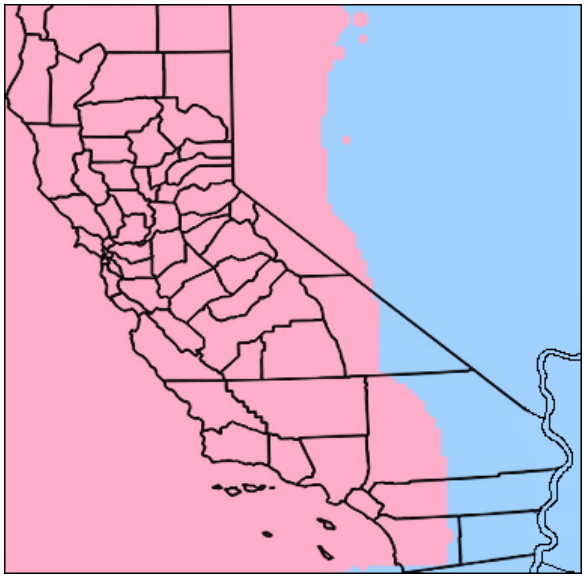

Figure 3: California precipitation cluster map (k means)

We see a somewhat skewed dataset, with a solid majority of data points located in cluster A, encompassing the ocean, most of the coast, and much of the mainland as well. Given this, and the fact that cluster B mostly encompasses the Nevada desert, it should come as no surprise that it has a higher average precipitation, and clearly a greater longitude. The latitude difference is much less, of course, which is to be expected given that both cover the entire vertical range, just about evenly.

For hierarchical clustering, we also get a two-way split, however this time it's horizontal rather than vertical, and the boundary between the two is rather less even:

Cluster	Avg. Precipitation	# Power Plants
A	-0.2265	232
B	-0.3786	548

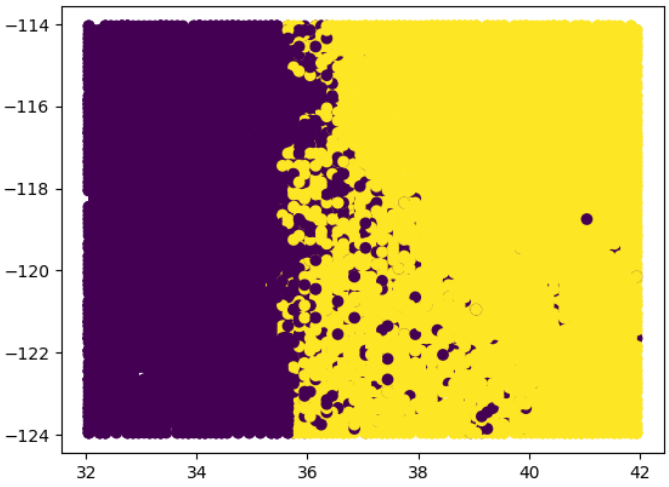

Figure 4: Precipitation clusters (hierarchical)

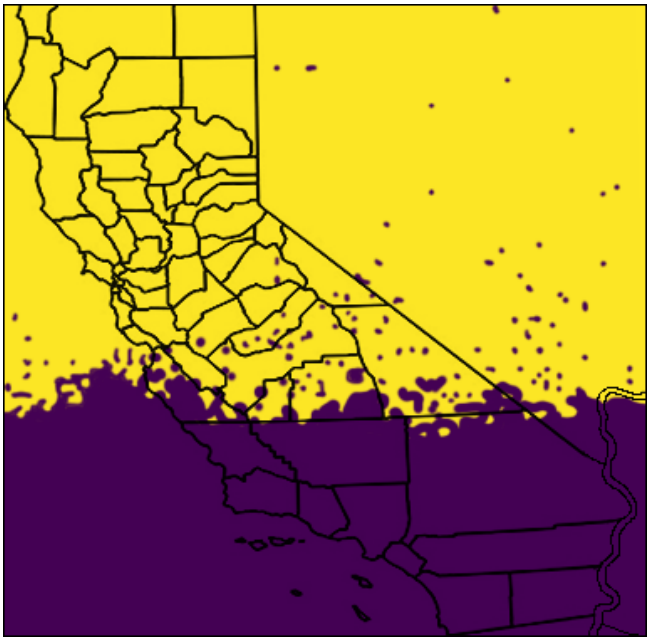

Figure 5: California precipitation cluster map (hierarchical)

This split does an equally good job describing the precipitation. It makes sense that the upper half would be a little wetter, and the lower half, cluster B, having more power plants does make sense, given that it contains almost all of the Colorado river.

Dataset	Type	Distance metric	Score
Precipitation	k-means	Euclidean	0.5897…
Precipitation	k-means	Manhattan	0.5897…
Precipitation	hierarchical	Euclidean	0.3199…
Precipitation	hierarchical	Manhattan	0.3443…

Evaluating the two models, we can see that, regardless of the distance metric, k-means performed better than hierarchical clustering. Interestingly, while changing the distance metric barely affected the performance of k-means, the hierarchical model did noticeably better when switching to Manhattan. The next dataset we look at is our soil moisture data. Here, k-means gives us another clean split:

Centroid	Soil Moisture	Availability %	Surface Temp	# Power Plants
A	0.5859	0.5793	0.5781	298
B	0.5719	0.5778	0.5788	464
C	-1.7281	-1.7284	-1.7284	18

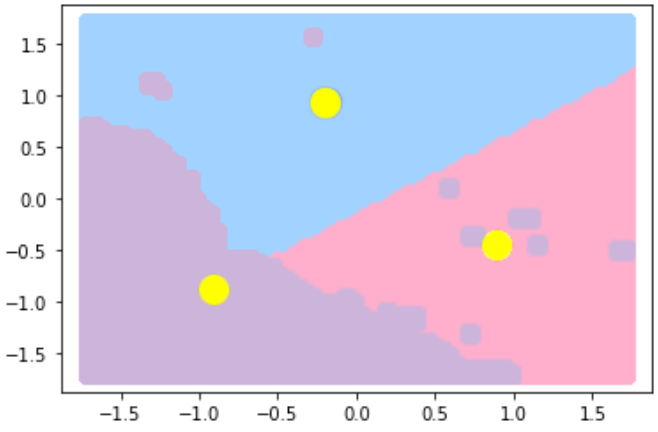

Figure 6: Soil moisture centroids (k means)

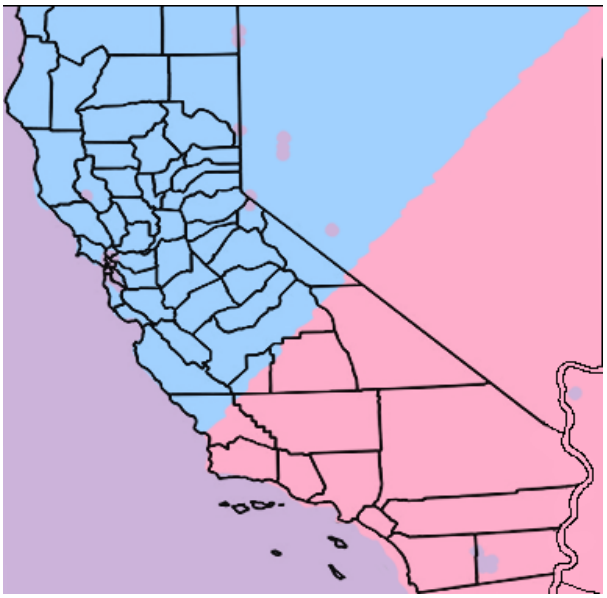

Figure 7: Soil moisture cluster map (k means)

Despite being the dataset with the most features per data point, we see the simplest distribution of clusters, with half of the landmass going to clusters A and B, and the ocean belonging to cluster C. The soil moisture values in the ocean approach negative infinity, so it makes sense that they would be in a class of their own compared to actual values, such as one can get on land. Interestingly, our values for A and B are rather similar, which makes sense given that we effectively divide the land in two even halves, clearly separating on latitude and longitude more than anything else. Given this divide, one should expect more power plants in cluster A, due to the lower coast being more populous since it contains Los Angeles and San Diego, as well as the Colorado river, as discussed previously. Amusingly, there do appear to be some power plants in cluster C, the "ocean cluster", though these are certainly due to the small sections of genuine coast in the cluster, as well as the several inland patches the cluster possesses.This time, the hierarchical clustering is closer to that given by k-means, though again much rougher:

Cluster	Soil Moisture	Availability %	Surface Temp	# Power Plants
A	-1.7344	-1.7347	-1.7347	19
B	0.5806	0.5869	0.5760	298
C	0.5687	0.5756	-0.5774	463

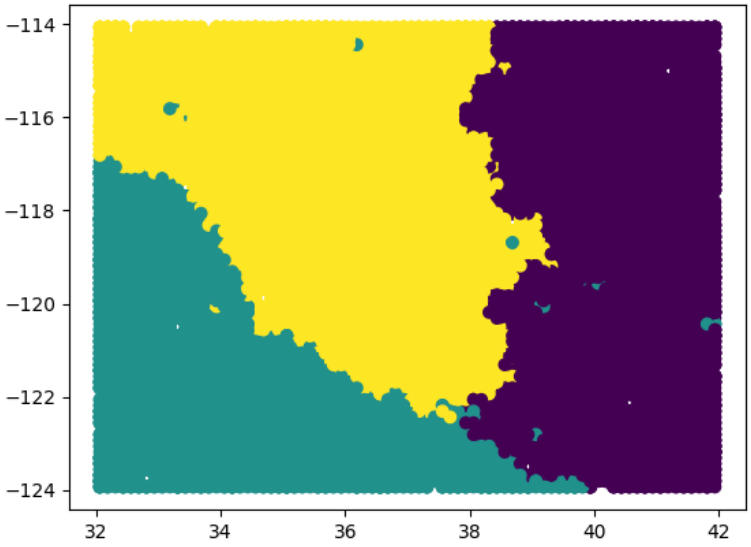

Figure 8: Soil moisture clusters (hierarchical)

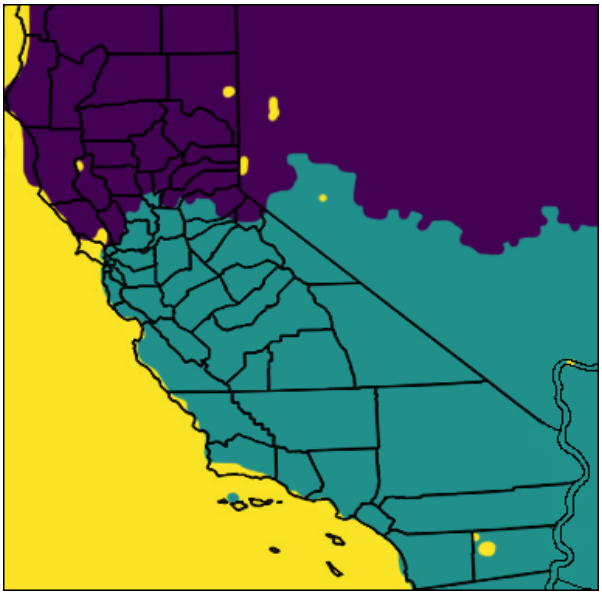

Figure 9: Soil moisture cluster map (k means)

Much like with k-means, we see the ocean mostly getting its own cluster, with the rest of the land split between the other two clusters. This time, cluster C seems to have gotten most of the land, with most of the power plants along with it, as one would expect. The real difference between the clusters is the ocean cluster vs the two land clusters, and other than that large difference with A, clusters B and C seem largely similar, although the surface temperature is a little lower in cluster C, the northern half of California, which makes sense.

Dataset	Type	Distance metric	Score
Soil Moisture	k-means	Euclidean	0.1055…
Soil Moisture	k-means	Manhattan	0.1055…
Soil Moisture	hierarchical	Euclidean	0.2954…
Soil Moisture	hierarchical	Manhattan	0.2994…

This time, hierarchical clustering performed better. K-means clustering again performed identically regardless of the metric, and hierarchical clustering again benefited from using Manhattan distance, however this time hierarchical performed noticeably better. Looking at the map, it does seem more logical to split the land area of California into north/south rather than diagonally 50/50, so I suppose it's good that we can justify this mathematically. Our final dataset is solar radiation:

Centroid	Solar Radiation	# Power Plants
A	0.6315	399
B	0.5016	363
C	-0.9722	18

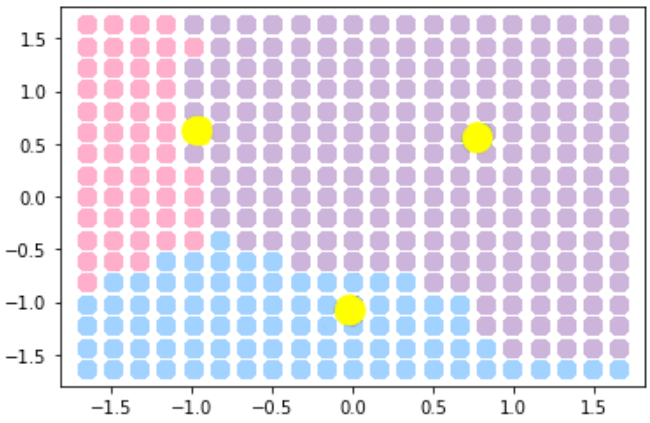

Figure 10: Solar radiation clusters (k-means)

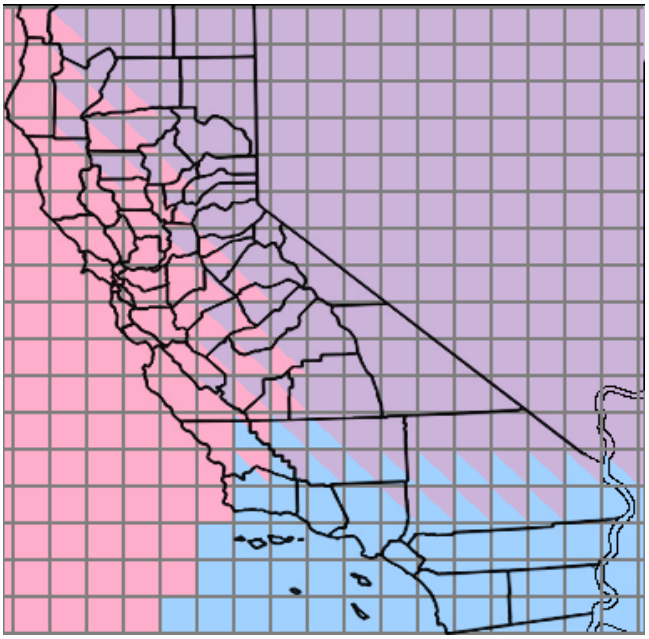

Figure 11: Solar radiation cluster map (k means)

There were much fewer solar radiation data points, spaced evenly in a grid. Where cluster assignments overlap, due to data points going back as far as 1980, that grid square is shown with the colors of both clusters. Luckily, there were no points labeled as belonging to all three clusters. We see more solar radiation in clusters A and B than C, which is likely explained by them sharing the coast. Interestingly, while there is clear "competition" for points along the borders of cluster C, there is not such great competition between A and B, probably due to the

greater difficulty of collecting data on the ocean, where most of their border lies. They also have nearly the same number of power plants, as unlike in the precipitation clusters, cluster B does not have the entire Colorado. If we look at what hierarchical clustering gets us, we see a very similar clustering to what k-means gives:

Cluster	Solar Radiation	# Power Plants
A	0.6315	42
B	0.5016	364
C	-0.9722	374

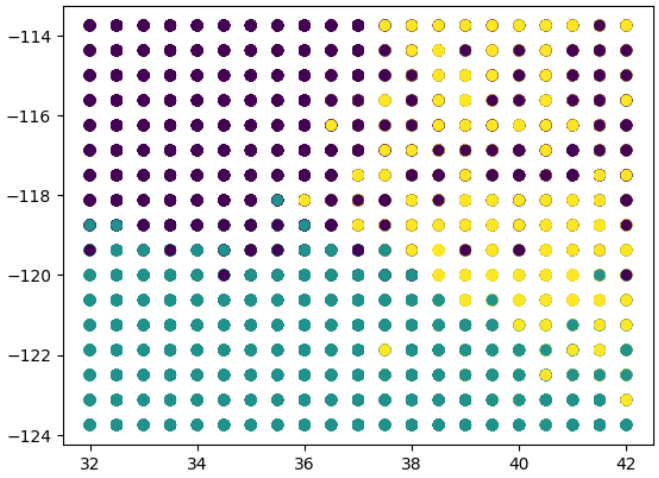

Figure 12: Solar radiation clusters (hierarchical)

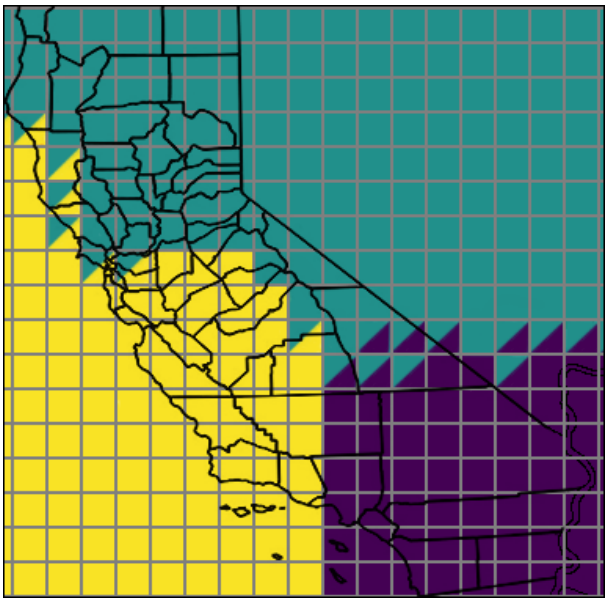

Figure 13: Solar radiation cluster map (k means)

It seems that with each dataset, our hierarchical clustering map grows ever more similar to that given by k-means. We have the same basic setup, dividing the map into rough thirds, with a little less solar radiation in the southeast, and the other two clusters having roughly equal solar radiation. Since, as mentioned previously, our map is now a grid, we have a much cleaner distribution, though I was a little surprised to not see any random pockets of some cluster contained within another like we usually see with hierarchical clustering.

Dataset	Type	Distance metric	Score
Solar Radiation	k-means	Euclidean	0.1066…
Solar Radiation	k-means	Manhattan	0.1066…
Solar Radiation	hierarchical	Euclidean	0.2145…
Solar Radiation	hierarchical	Manhattan	0.2243…

Again, it seems that hierarchical clustering has outperformed k-means. We once again see the familiar pattern of k-means being unaffected by our distance metrics, with hierarchical clustering benefitting from Manhattan, and in our tiebreaker round, hierarchical clustering is the victor.

5) It seems that overall, hierarchical clustering did a better job on our environmental datasets than k-means. I must admit, I did not expect to find a clear winner, thinking that given the same dataset, particularly ones without anything crazy going on like we find here, we would get very similar results between the two methods. Even without the mathematical interpretation, I find that the rougher, more organic maps it generates are more true to reality than the clean, crisp border lines often formed by k-means. One thing that I am not sure of, is why the distance metric affected the two methods so differently. It seems logical that if it wouldn't affect one then it wouldn't affect the other, and vice-versa. I would be interested in looking deeper into the methods to justify this discovery, however that is outside the scope of this project. One thing that I noticed was that hierarchical clustering ran much slower than k-means. It seems that k-means runs in polynomial time, whereas hierarchical clustering has a time complexity of O(n^3), which certainly explains this. The datasets each contained over a hundred thousand datapoints, which is absolutely large enough to notice the difference between the two runtimes. Were I to continue this project, I would of course be interested in examining different clustering methods, as well as seeing if there are other interesting measures of distance to use. One of the main other distance metrics is cosine distance, however this method is for when the magnitude of vectors doesn't matter, and I thought it ill-suited for this data. More importantly, there are many interesting methods of clustering out there, and I would like to see the different ways they cluster the data, and particularly the maps they make. For example, unlike the two methods I used, density based

clustering methods like DBSCAN ignore certain points as outliers, so I would be interested in seeing what gets considered as noise.

Github link: https://github.com/RyanKaplan999/DataAnalytics_A6_Ryan_Kaplan

References:

1. Frades, I. & Matthiesen, R. Overview on techniques in cluster analysis. Bioinforma. methods clinical research 81–107 (2010)

2. Shahapure, K. R. & Nicholas, C. Cluster quality analysis using silhouette score. In 2020 IEEE 7th international conference on data science and advanced analytics (DSAA), 747–748 (IEEE, 2020).

3. Murtagh, F. & Contreras, P. Algorithms for hierarchical clustering: an overview. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2 , 86–97 (2012).