# Classifying COVID Data using K-means Clustering


Ryan Kropp
Electrical and Systems Engineering Deparement
Washington University
St. Louis, United States of America
kropp.r@wustl.edu

Matthew Walker
Electrical and Systems Engineering Department
Washington University
St. Louis, United States of America
matthew.r.walker@wustl.edu

Robert Walsh
Electrical and Systems Engineering Department
Washington University
St. Louis, United States of America
r.j.walsh@wustl.edu



**In this project, we sought to find a clear method to classify COVID-19 data by region. The idea was to have a working method that could accurately determine the region of a county solely based on the input of new COVID-19 cases per capita each week. While a perfect method is always the goal,-we aimed to create a method that could cluster the data more accurately than just guessing alone. After running many different methods, including data amplification and region difference analysis, we finally decided on a diversity ratio method wherein we compared one region's standard deviation against every other region's standard deviation in order to find the timepoints where this particular region had the least data diversity, while the other simultaneously possessed larger data diversity. The intention was that of identifying timepoints where trends were *unique* to the region in question alone. This ratio was calculated for every region at every timepoint.**


## I. Introduction

For this project, the goal was to use our understanding of k-means clustering and matrix manipulation to reliably derive location information based only on COVID-19 data. We hoped to use this real-life scenario to apply and enrich our knowledge of linear algebra and find a means that could place counties into a geographical region based solely on the cases per capita each week. We aim to end this project with a method that can work at least twice as accurate as guessing, since a perfect classification method using this data alone is near impossible given there are a plethora of variables that cannot be accounted for via this data.

## II. Methods

Our goal for the project was to derive and execute a method which would extract geographical information from raw Covid-19 case count data. More specifically, we wanted to be able to identify a county's general location in the United States based on its Covid data alone.

### *A. The Data Set*

The data we examined was in the form of matrices, the most significant of which being a 225x130 matrix. Setting integer i as the first dimension and j as the second, the element (i, j) represented the ith county's #/covid cases/100,000 people for the jth week. Additional data consisted of a 225-element reference vector, which showed the region number for each county, as well as census data that gave each county's population, location, etc.

To begin, we simply wanted to generate visual representations of the data to see if there were any easily distinguishable trends:

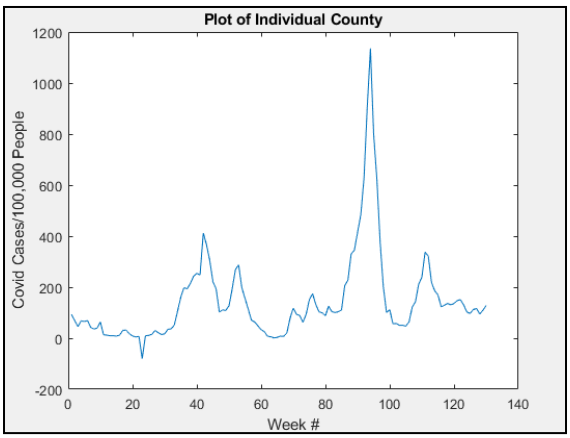


Figure A: Plot of an individual county, selected at random from region 1.

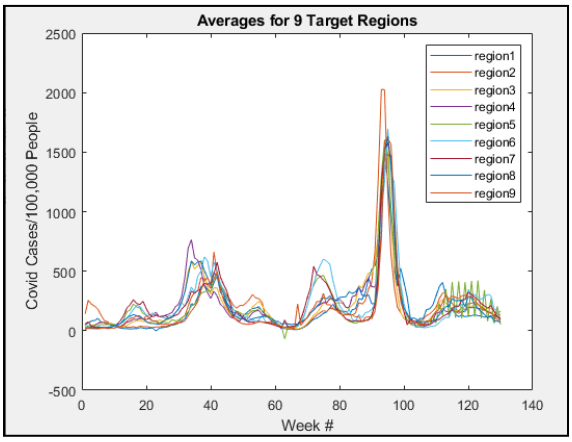


Figure B: Plot of the averages for each region, demonstrating how many significant timepoints (i.e. peaks) line up, though there are still identifiable differences.

After observing the plot of the averages, we decided we needed a way to find the most diverse timepoints, whereas few regions as possible contained similar data. The idea was that if we could isolate these timepoints, then we could utilize the kmeans method in conjunction with this selective diversity, and resultingly generate centroids that automatically drew counties from their respective regions to their own cluster.

### *B. The Math Behind the Method*

Initially, we thought to use the averages of the regions as a baseline. We wanted to look at the differences between every county for a region and that region's mean, when we realized there's a mathematical function that does this for us: the standard deviation. We intended to use standard deviation to determine the *diversity* of each region, relative to the diversity of every other region. The following is the equation by which we determined region i's **diversity ratio** at week j:

$$STDV(region_i(:,j))/SUM(STDV(region_k(:,j))). \quad (1)$$

Where k is every integer **not** equal to i, from 1-9. The colons indicate that every single datapoint of the 25 counties corresponding to a region are included in the STDV equation.

The rationale behind this calculation is this: If a particular region has a **low** deviation at timepoint j, that means that the region is relatively consistent at this timepoint. If the other regions, for the most part, have a **high** deviation at the same timepoint, then they are relatively inconsistent there. The result of dividing these two is a ratio, where a lower value indicates that the time point corresponding to it is a good marker index for that region, given the region in question is consistent while most others are not.

The minimum values found via this method would be at the best marker indices. Constraining the dataset to these, in theory, would provide better focus to the auto-centroid generation of MATLAB's "kmeans" method, which we used in the next step.

### *C. Method in the Code*

All of our number handling/manipulation was done in MATLAB's 2022a version. To begin we imported the relevant data provided, and then the first few sections of our code consisted of simple matrix slicing and concatenating("cat" in MATLAB.) These we used to create our "region" and "average" matrices.

Once we had our sliced matrices, we plotted what we found useful to visualize. After came implementing our method, which mostly involved loops to run through, examine, and augment all of our region data.

First, we used MATLAB's "std" function to calculate the standard deviation of every timepoint for each region. These we concatenated into a 9x130 matrix, and this was run through a nested loop that automatically calculated and appended the "diversity ratio" to its proper index in a new 9x130 matrix. MATLAB's "mink" function found the smallest 40 ratios for each region, which we compiled. These values were checked to ensure the corresponding indices weren't good markers for too many regions, as this would effectively make them bad markers.

Once these values were pulled, an "odd identity matrix" was produced, in which the main diagonal only held ones at the marker indices pulled from the last step, i.e. oddID(i,i) = 1 if i was selected as a good marker index.

The county covid data matrix was multiplied by this 130x130 odd identity matrix, which resulted in all columns' deletion excepting the marker columns. We split the result up into a sample data set and a test data set, the former used to generate centroids via kmeans and the latter to run through kmeans using those centroids as a baseline. Additional loops determined which region each centroid corresponded to, as well as how many of the test counties landed in their proper region according to their cluster.

## III. Results and Discussion

For our results we received a 60% accuracy rate through the means of using train and test data for our k-means.

In figure 1 we can see a silhouette chart of our test data after it has been processed with k-means. Based on the chart here, we can see harsh negative values for many of our clusters which indicates that k-means believes that the data points are so like one another that nine clusters may not have been the best choice for k despite there only being nine regions.

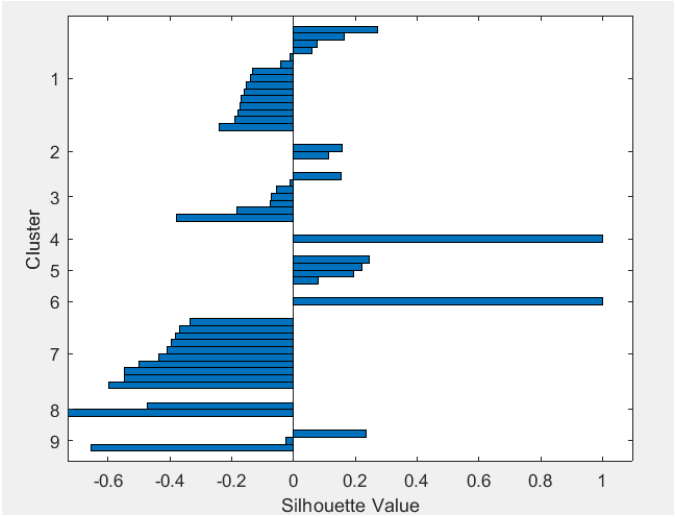


Figure C: Silhouette Values of Clusters

Our work suffered from a few limitations from our data. The data had inconsistencies, with some data points having an increasing and decreasing infection rate even within the same regions and divisions. These limitations seem to be caused by the data being measured at different points of outbreak across the United States. A certain area may have already seen the brunt of the pandemic and is experiencing a decline in infections, while other areas in the same region may have a ramping up infection rate due to Covid just reaching them. This alone makes it very tricky to determine the region of a county based on infection rate alone.

## IV. Conclusion

We discovered through this case study that correctly sorting data into clusters is not as simple as running a basic k-means algorithm. Most data sets, especially ones like this Covid set, require a significant pre-processing period before it can be properly sorted and assigned clusters. Through our toil we managed to receive a 60% accuracy rate. With this accuracy rate, our shortcomings are apparent, but this parallels the overall difficulty for health officials to process the data in the real world to help combat the pandemic.

We believe we can expand upon our findings by processing additional data, and rather than trying to determine each region, we could use k-means to determine the most at-risk locations for disease breakouts.

## References


[1] Boyd, S., & Vandenberghe, L. (2018). *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. Cambridge: Cambridge University Press. doi:10.1017/9781108583664

[2] Bureau, US Census. "County Population Totals: 2020-2021." *Census.gov*, 1 Mar. 2022, https://www.census.gov/data/datasets/time-series/demo/popest/2020s-counties-total.html.

[3] "MATLAB Deep Learning Toolbox." *Deep Learning Toolbox Documentation*, https://www.mathworks.com/help/deeplearning/index.html.

[4] The New York Times. "Coronavirus (Covid-19) Data in the United States." *GitHub*, 2021, https://github.com/nytimes/covid-19-data.