# Final_Exam

Andrew Gutierrez

2022-12-18

For my final exam submission, I'll be utilizing a publicly-available dataset published by Lake County, Illinois through catalog.data.gov. The dataset shows the cancer rates per 100,000 people for the 27 ZIP codes that are present in the county, with further breakouts by type of cancer (Colorectal, Lung, Breast, Prostate, and Urinary cancers).

For this project, I'll be doing an analysis clustering the 27 ZIP codes by their cancer rates in order to visualize and understand the varying rates of propensity for cancer among the different populations within Lake County. I'll then use that model to entertain a hypothetical scenario: suppose Lake County decided to annex a previously-unincorporated community just outside the county borders, creating a new ZIP code for the community in the process. How would this new community fit into the pre-existing clusters based on cancer rates?

First, I'll install the requisite packages.

```
library(factoextra)
```

```
## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(flexclust)
```

```
## Loading required package: grid

## Loading required package: lattice

## Loading required package: modeltools

## Loading required package: stats4
```

```
library(clue)
```

```
## Warning: package 'clue' was built under R version 4.2.2
```

Next, I'll read the CSV file into a DataFrame in R:

```
CR = read.csv("C:\\Users\\gutiera9\\Documents\\MSBA KSU\\LakeCounty_Illinois_CancerRates.csv",header=T,
```

I'll do some brief manipulation of the DataFrame here, dropping the "FID", "Shape Length", and "Shape Analysis" columns which will not factor into our analysis, and also converting the ZIP code column to a "character" data type so that it is not read as an integer.

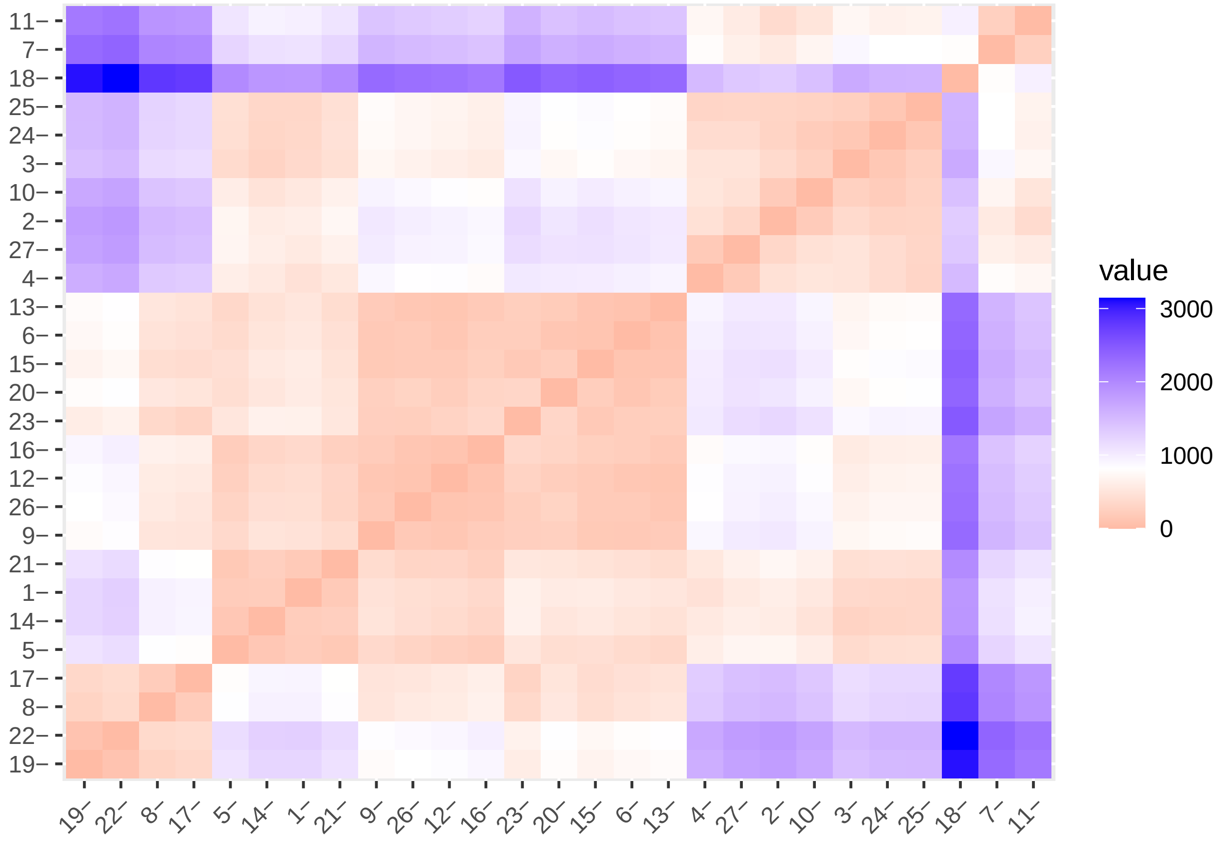
```
CR <- CR[-c(1,9:10)] # drop non-relevant columns

CR[c(1)] <- lapply(CR[c(1)], as.character) # convert ZIP column to character

head(CR)
```

```
##     ZIP Colorectal Lung_Bronc Breast_Can Prostate_C Urinary_Sy All_Cancer
## 1 60002   218.0621   419.6667   399.0948   259.2059   259.2059   2703.148
## 2 60010   258.9157   335.4647   504.3228   499.8199   227.3955   3248.829
## 3 60015   153.4359   230.1538   478.5738   442.0414   222.8473   2922.588
## 4 60020   292.7972   507.5151   214.7179   302.5571   370.8764   3084.130
## 5 60030   221.5354   284.4406   404.7808   322.7306   210.5954   2581.845
## 6 60031   163.5021   221.5190   414.0295   303.2700   160.8650   2217.827
```

Fortuitously, since all of the numerical variables in this dataset are on a “cases per 100,000 persons” scale, none of the variables will need to be normalized.

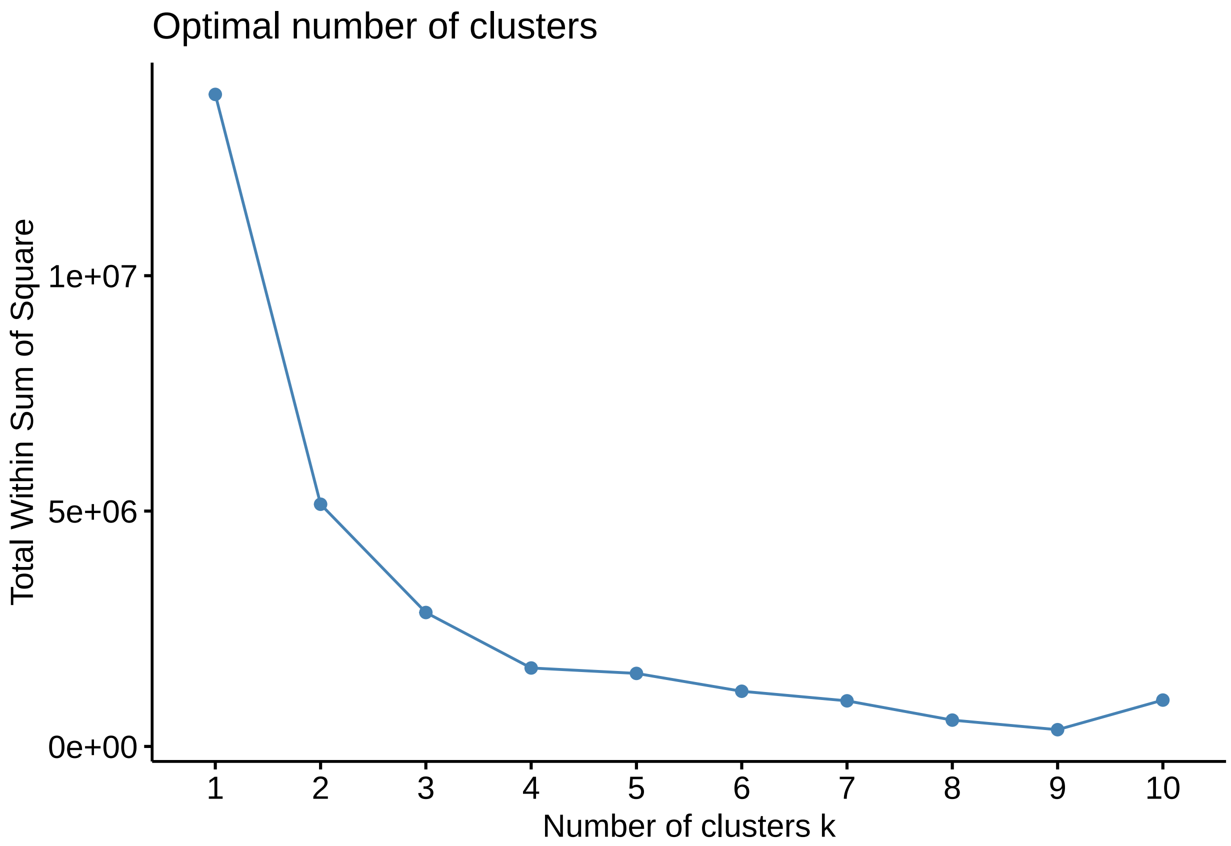
Next, I’ll calculate the Euclidean distance between the 27 ZIP code records in the dataset. I chose Euclidean distance because A) none of the numerical variables in the dataset have any extreme outliers (the cancer rates here all fall within a pretty well-defined range), and B) the variables are all on the same scale.

```
distance <- get_dist(CR[c(2:7)],method="euclidean") # calculate distances
fviz_dist(distance) # display the distances in a graph
```



Next, I’ll determine the optimal value of k (i.e. the optimal number of clusters) using an elbow chart.

```r
fviz_nbclust(CR[c(2:7)], kmeans, method = "wss")
```



To my eye, the clear "elbow bend" on the above chart that affords the best balance between model bias and overfitting is k = 3, as WSS begins decreasing at a much smaller rate beyond that point.

I'll now run my clustering model using the k-means algorithm, as it works particularly well when used with Euclidean distances.

```r
k3 <- kmeans(CR[c(2:7)], centers = 3, nstart = 25) # run k-means algorithm
```

Let's take a look at the number of ZIP codes within each cluster, as well as the cluster centers.

```r
print(k3$size) # print the size of each cluster
```

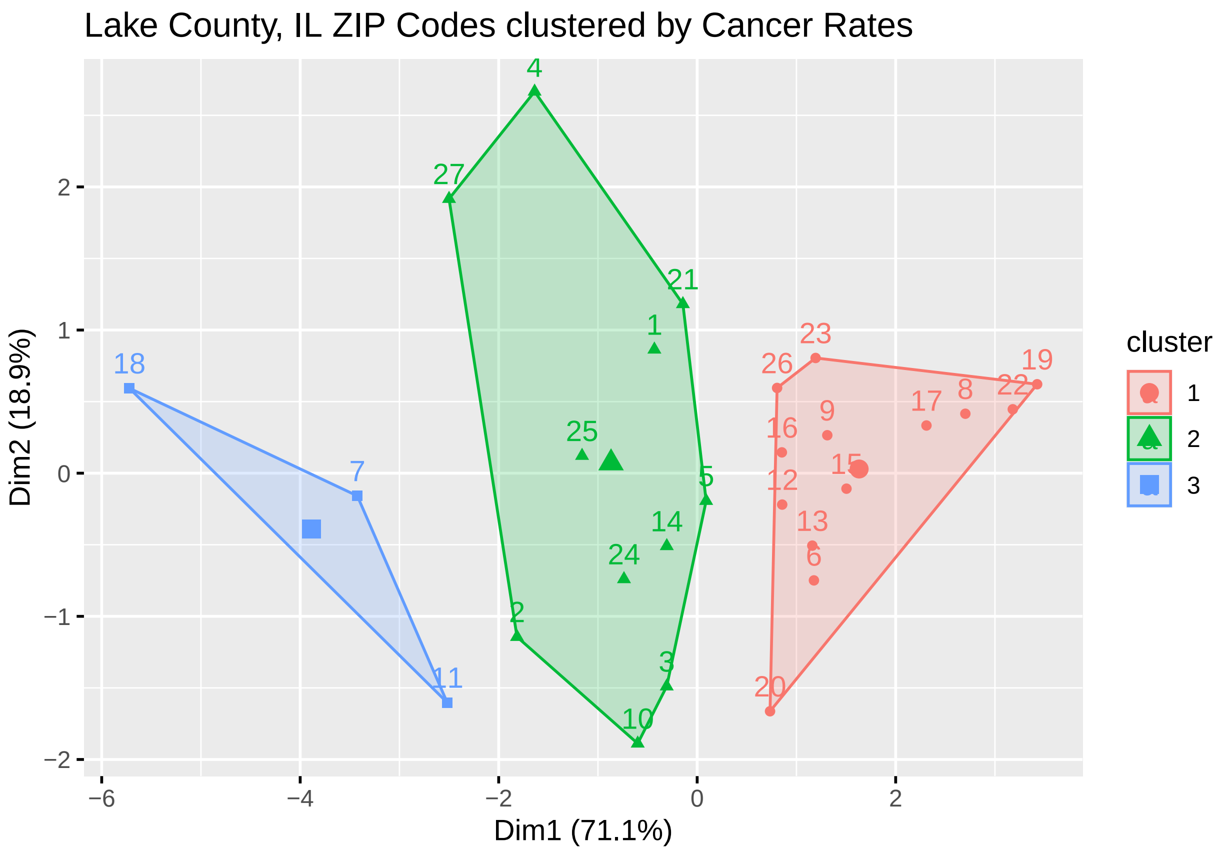
```
## [1] 13 11  3
```

```r
print(k3$centers) # print the centers of each cluster
```

```
##   Colorectal Lung_Bronc Breast_Can Prostate_C Urinary_Sy All_Cancer
## 1   166.6786   237.5092   333.2354   245.7300   163.9990   2070.266
## 2   245.3301   344.6806   407.8148   364.5543   251.8827   2924.759
## 3   319.1408   456.9962   596.2386   479.1905   365.4909   3959.243
```

And now, let's take a look at the clusters, visualized:

```r
fviz_cluster(k3, data = CR[c(2:7)], show.clust.cent=TRUE,main='Lake County, IL ZIP Codes clustered by Ca
```



What this visual and the preceding stats show are three distinct clusters:

- Cluster 1, "Low-Cancer Rate ZIP Codes". This cluster contains 13 ZIP codes and has a cluster center that represents an overall cancer rate of 2,070 cases per 100,000 persons.
- Cluster 2 "Medium-Cancer Rate ZIP Codes". This cluster contains 11 ZIP codes and has a cluster center that represents an overall cancer rate of 2,924 cases per 100,000 persons.
- Cluster 3 "High-Cancer Rate ZIP Codes". This cluster contains 3 ZIP codes and has a cluster center that represents an overall cancer rate of 3,959 cases per 100,000 persons.

I'll now merge the cluster output from my model with the original dataframe - allowing me to easily see which ZIP codes have been allocated to which clusters.

```r
CR <- cbind(CR,data.frame(k3$cluster))
print(CR[,c(1,7:8)])
```

```
##       ZIP All_Cancer k3.cluster
## 1  60002   2703.148          2
## 2  60010   3248.829          2
## 3  60015   2922.588          2
## 4  60020   3084.130          2
## 5  60030   2581.845          2
## 6  60031   2217.827          1
## 7  60035   3760.432          3
## 8  60040   1796.296          1
## 9  60042   2267.415          1
## 10 60044   3149.850          2
## 11 60045   3611.815          3
## 12 60046   2340.859          1
## 13 60047   2261.908          1
## 14 60048   2697.719          2
## 15 60060   2174.085          1
## 16 60061   2409.731          1
## 17 60064   1830.419          1
## 18 60069   4505.481          3
## 19 60073   1533.541          1
## 20 60083   2205.741          1
## 21 60084   2596.584          2
## 22 60085   1465.294          1
## 23 60087   2092.970          1
## 24 60089   2991.535          2
## 25 60096   2995.658          2
## 26 60099   2317.369          1
## 27 60041   3200.462          2
```

Now that my model is complete, I'll entertain my hypothetical scenario - the expansion of Lake County, Illinois. How would our new ZIP code addition fall within these clusters?

I'll start by creating a new dataframe containing only the new data point.

```
NewZIP = data.frame(ZIP = "60100", Colorectal = 270, Lung_Bronc = 150, Breast_Can = 250 , Prostate_C = 3

NewZIP
```

```
##     ZIP Colorectal Lung_Bronc Breast_Can Prostate_C Urinary_Sy All_Cancer
## 1 60100        270        150        250        350        275       3010
```

And I'll now use the existing k-means model to make my prediction.

```
cl_predict(k3, NewZIP[c(2:7)])
```

```
## Class ids:
## [1] 2
```

Running the new data point through the model shows that the new annexation to Lake County comfortably fits within the “Medium-Cancer Rate ZIP Codes” cluster.