K-Means Clustering Analysis on the ECOLI Dataset

Theodore Fitch
Department of Data Analytics, University of Maryland Global Campus

DATA 630: Machine Learning

Dr. Ami Gates

July 31st, Summer 2021

Introduction:

Protein localization is the sequestering of proteins into a specific area of the cell in order for critical functions to exist (Kaeberlein and Kennedy, 2007). The cell and by extension all organisms run by proteins. Where DNA (Deoxyribonucleic acid) contains all of the information for an organism to exist, proteins are how the animal, plant, or cell actually exists and gets anything done. Where DNA is the blueprint, protein is the building. DNA will be transcribed to mRNA (or “messenger ribonucleic acid”) in the center of the cell (Kaeberlein and Kennedy, 2007). This mRNA will leave the center of the cell thus protecting the DNA from any harmful processes (because it stays safe in a shell call the nucleosome and because if the mRNA becomes corrupted, the original DNA still exists from which another copy can be made).

Proteins are made of repeating units of amino acids of which there 20 commonly occurring (with many variants within those basic types of amino acids)(Weber and Miller, 1981). Each amino acid has its own unique flavor – some are hydrophobic (do not like water) and some are hydrophilic (they like water). This difference in the amino acids will cause proteins to fold up (like a ball of string scrunched together) in such a way where the hydrophobic ones will all hide together in the center and the hydrophilic ones will end up on the outside. This is because it is the most energetically favorable way for the protein to exist so it will naturally fold up that way (Weber and Miller, 1981). Based on the way each protein folds, it will have different functions. Some proteins help other proteins get made. Some proteins create the “skeleton” of the cell. Some proteins help get rid of toxins or kill viruses inside the cell. Proteins are ubiquitous throughout cells because so many exist because they each a unique function (Kaeberlein and Kennedy, 2007). Protein is the reason for life.

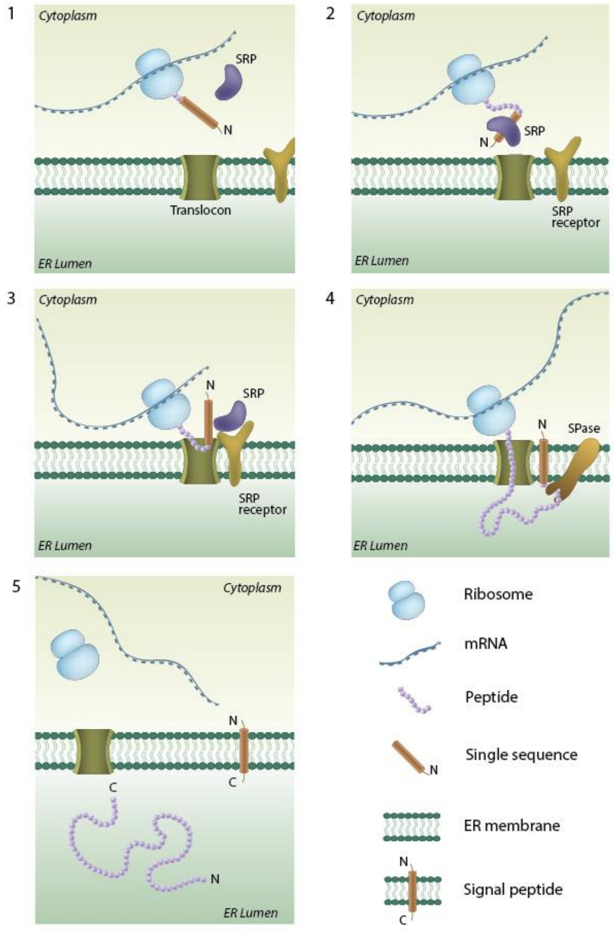


Figure 1. Protein transcription is essential to understand in order to understand protein localization. (MBInfo, n.d.).

Thus, where the proteins sequester is important because many are needed for their function. Many will be made at once in order for them to fulfill their function. There are a variety of ways the proteins arrive to their different sites. They may be able to passively dissolve to their site where they are needed. But more commonly they are “tagged” with a signal sequence to highlight them to transfer proteins. These transfer proteins will grab them and travel along the cytoskeleton (the actin matrix inside the cell). This will move the proteins from one region to another. However, predicting where the proteins will gather is critical. They may be in the cytoplasm (free-floating in the inner matrix of the cell). They may be attached to the inside or outside of the phospholipid bilayer (which acts as the barrier of the cell). They may be in the periplasm (which is the space between the inner phospholipid bilayer and the outer phospholipid bilayer (two-exist in the gram-negative bacteria)(MBInfo, n.d.). It is critical to know where a protein localizes to understand its purpose, to watch it work in real time, and in some cases to harvest the protein from the cell.

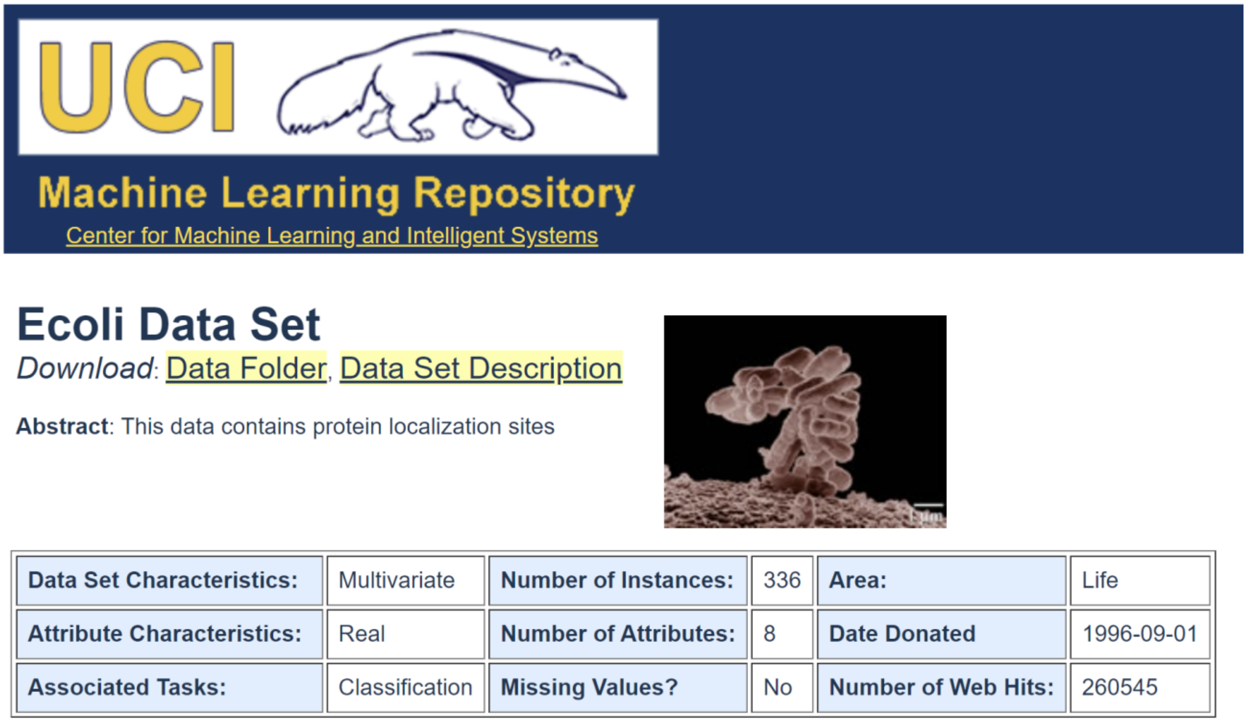
It has been past practice to predict protein localization sites manually by assessing the amino acid sequence and finding homogenous sequences. This entails looking at the amino sequences and comparing those sequences to known proteins which have been assessed in the past and whose functions are known. However, this problem is particularly effective for machine learning methods to solve because it involves taking a known sequence and comparing it to a very high volume of other sequences looking for a certain level of similarity.

When attempting to understand the cell, Escherichia coli, or E. coli for short, has been used as the prime example (Berg, 2008). It is incredibly simple and yet incredibly complex. It has plenty of biochemical pathways which with humans share homogeny. It is a hearty organism easy to grow thousands of colonies (made up of thousands of cells) within a very short timeline. Thus, it is also easy to study mutations and evolution because it grows its generations so quickly. E. coli is one of the most well studied organisms in all of nature. Because of this, it can also be used to understand protein localization reflexively. It is well documented that the protein sequences of E. coli can be used to predict their localization sites (Sjöström et al., 1987).

Analysis and Model Demonstration:

Data Information:

This dataset was procured from the UCI website (UCI Machine Learning Lab, 1996). The dataset was generated from real-world data. It was in CSV format. It shall be referred to as "ECOLI" henceforth for the sake of brevity. ECOLI was originally created in an attempt to predict localization sites of proteins within the cell.


UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Ecoli Data Set

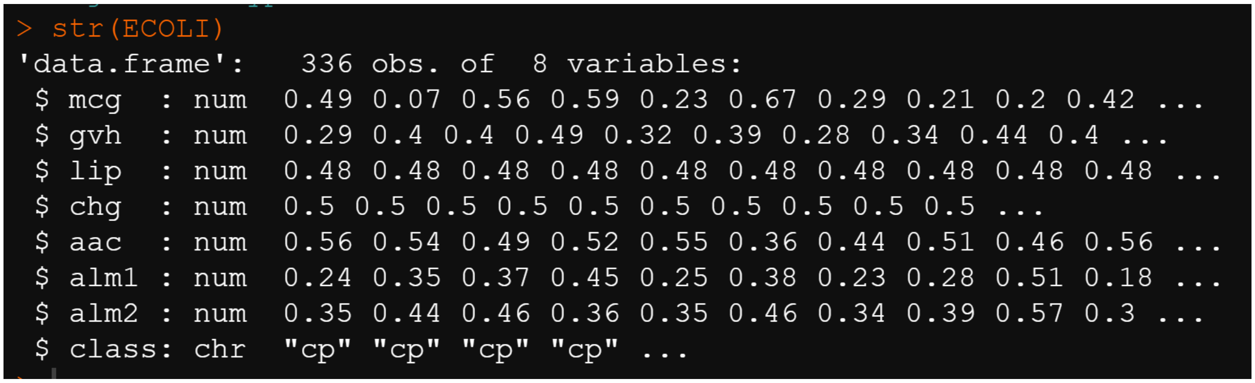
Download: Data Folder, Data Set Description

Abstract: This data contains protein localization sites

Data Set Characteristics:	Multivariate	**Number of Instances:**	336	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	8	**Date Donated**	1996-09-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	260545

Figure 2. The UCI website shows the summary of the ECOLI Data Set.

Exploratory Data Analysis:



```
> str(ECOLI)
'data.frame':   336 obs. of  8 variables:
 $ mcg  : num  0.49 0.07 0.56 0.59 0.23 0.67 0.29 0.21 0.2 0.42 ...
 $ gvh  : num  0.29 0.4 0.4 0.49 0.32 0.39 0.28 0.34 0.44 0.4 ...
 $ lip  : num  0.48 0.48 0.48 0.48 0.48 0.48 0.48 0.48 0.48 0.48 ...
 $ chg  : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
 $ aac  : num  0.56 0.54 0.49 0.52 0.55 0.36 0.44 0.51 0.46 0.56 ...
 $ alm1 : num  0.24 0.35 0.37 0.45 0.25 0.38 0.23 0.28 0.51 0.18 ...
 $ alm2 : num  0.35 0.44 0.46 0.36 0.35 0.46 0.34 0.39 0.57 0.3 ...
 $ class: chr  "cp" "cp" "cp" "cp" ...
```

Figure 3. Using the command Structure on ECOLI showed there are 8 variables and 336 rows of data (only 7 variables are used for prediction and 1 is used to be predicted [class]).

Index	Variable	Meaning
1	mcg	McGeoch's method for signal sequence recognition
2	gvh	von Heijne's method for signal sequence recognition
3	lip	von Heijne's Signal Peptidase II consensus sequence score (binary attribute)
4	chg	Presence of charge on N-terminus of predicted lipoproteins (binary attribute)
5	aac	Score of discriminant analysis of the amino acid content of outer membrane and periplasmic proteins
6	alm1	Score of the ALOM membrane spanning region prediction program
7	alm2	Score of ALOM program after excluding putative cleavable signal regions from the sequence
8	class	Type of localization site. See Table 2

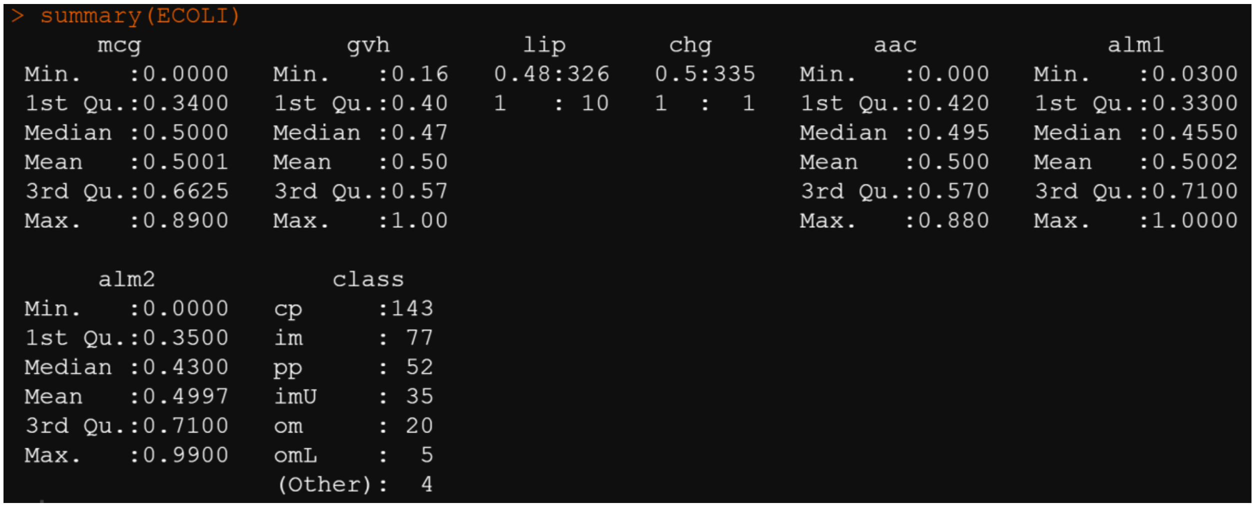
Table 1. There are 7 variables in the ECOLI dataset with 1 class variable to be predicted.

Index	Class	Number of Instances	Name
1	cp	143	Cytoplasm
2	im	77	Inner membrane without signal sequence
3	pp	52	Periplasm
4	imU	35	Inner membrane, uncleavable signal sequence
5	om	20	Outer membrane
6	omL	5	Outer membrane lipoprotein
7	imL	2	Inner membrane lipoprotein
8	imS	2	Inner membrane, cleavable signal sequence

Table 2. There are 8 classes in the ECOLI dataset listed above in descending order for number of instances.

ECOLI contained 8 variables with 336 rows of data. “Class” was the variable of interest which signified the localization site. Specifically, the names used are the accession number for the SWISS-PROT database. All other variables were numeric in nature. However, it is critical to note that lip and chg were binary variables (with values of 0.48/1 and 0.5/1 respectively). There were only 10 values of lip being 1 and there was only 1 value of chg being 1 (this value of chg was also one of the values where lip was 1).

A thorough discussion of each variable was created by Horton and Nakai (1996). In short, each variable describes a quantitative metric (or binary metric in the case of “lip” and “chg”) of identifying a key aspect of a protein. Some of these metrics describe the tags on proteins previously discussed. For instance, “chg” represents whether a charge is present on the N-terminus of a lipoprotein (Table 1). The von Heijne methods both revolve around predicting the signal sequence (the part of the protein actually used for its function) and the N-terminus (1986). When all of these attributes are analyzed, the model should attempt to cluster the values based on the “class” variable of where proteins are likely to localize.



```
> summary(ECOLI)
      mcg              gvh           lip        chg           aac              alm1       
 Min.   :0.0000   Min.   :0.16   0.48:326   0.5:335   Min.   :0.000   Min.   :0.0300  
 1st Qu.:0.3400   1st Qu.:0.40   1   : 10   1  :  1   1st Qu.:0.420   1st Qu.:0.3300  
 Median :0.5000   Median :0.47                        Median :0.495   Median :0.4550  
 Mean   :0.5001   Mean   :0.50                        Mean   :0.500   Mean   :0.5002  
 3rd Qu.:0.6625   3rd Qu.:0.57                        3rd Qu.:0.570   3rd Qu.:0.7100  
 Max.   :0.8900   Max.   :1.00                        Max.   :0.880   Max.   :1.0000  
                                                                                     
      alm2             class    
 Min.   :0.0000   cp      :143  
 1st Qu.:0.3500   im      : 77  
 Median :0.4300   pp      : 52  
 Mean   :0.4997   imU     : 35  
 3rd Qu.:0.7100   om      : 20  
 Max.   :0.9900   omL     :  5  
                  (Other):  4  
```

Figure 4. The summary command shows the distributions for the values across all 8 variables (graphic made with lip, chg, and class variables converted to factors. Those variables were converted back to integers after summary command was run).

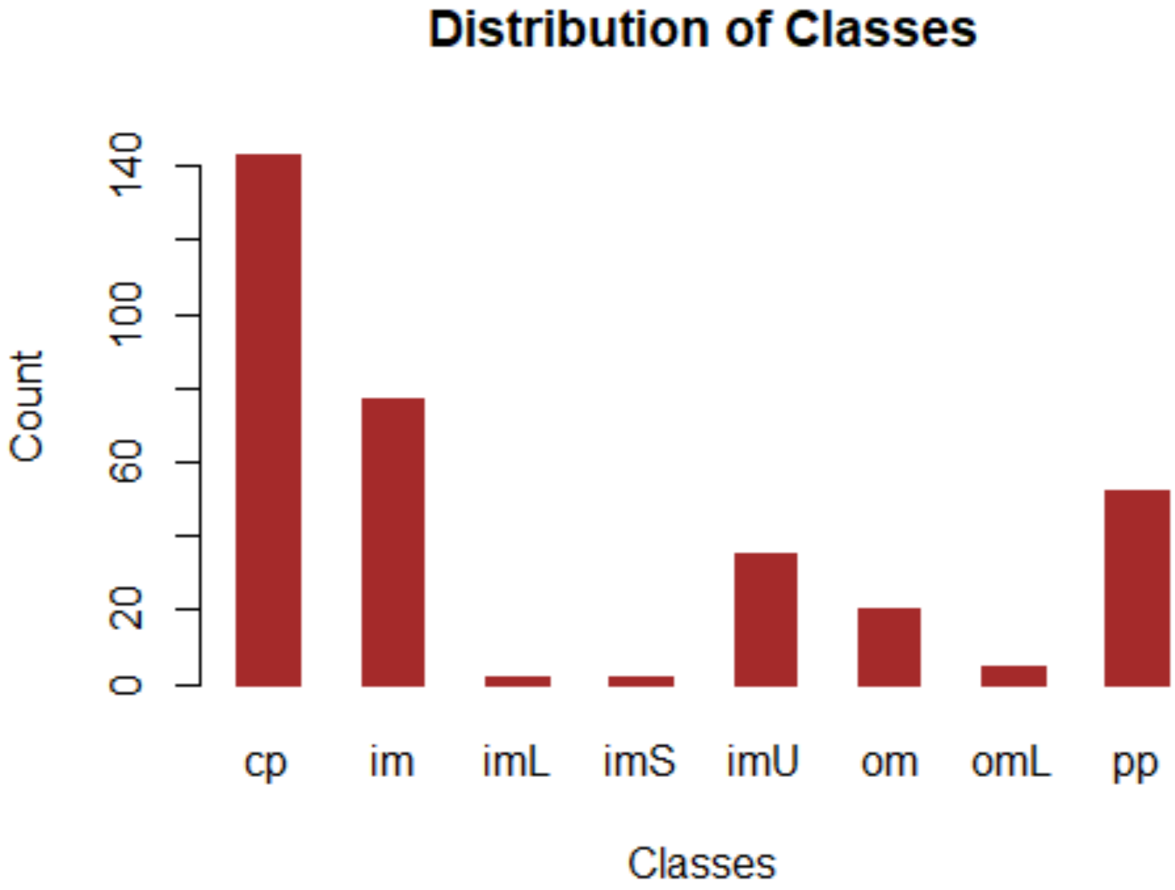


Figure 5. Distribution of the Class variable shows there were highly varying numbers of each class. 2 variables (cp and im) account for over 65% of the values (220 of 336).

Preprocessing:

ECOLI required minimal data preprocessing. There were no nulls detected. There were no outliers observed. "Class" was removed as the key at a few pertinent steps. All variables appeared to be relevant, and all rows appeared to be valuable at first exploration.

K-Means Clustering Method:

K-Means Clustering is one of the most popular unsupervised machine learning methods. Unsupervised learning specifically denotes machine learning programs that don't require labelled data (Han et al., 2011). Methods like regression require that the user enter the type of data that each variable is. This is because the method relies on knowing the types of data in order to make predictions. The method must know what the variable of interest is at minimum (which requires labelling). On the other hand, unsupervised methods aren't attempting to predict a particular variable. Instead, K-Means Clustering is looking for patterns within the data. The method requires that the input values are scaled and that the user inputs how many "clusters" are required. Then, it groups the values together based on similar patterns it detects into the specified number of clusters. There is no single way of determining how many clusters should be used. Each problem presented will demand a different number of clusters. Specific methods will be discussed soon. Once values are clustered together, a few analyses can be performed to observe what patterns the method detected. Clustering is used commonly to explore datasets, to extract features, and to extract novel variables (Han et al., 2011).

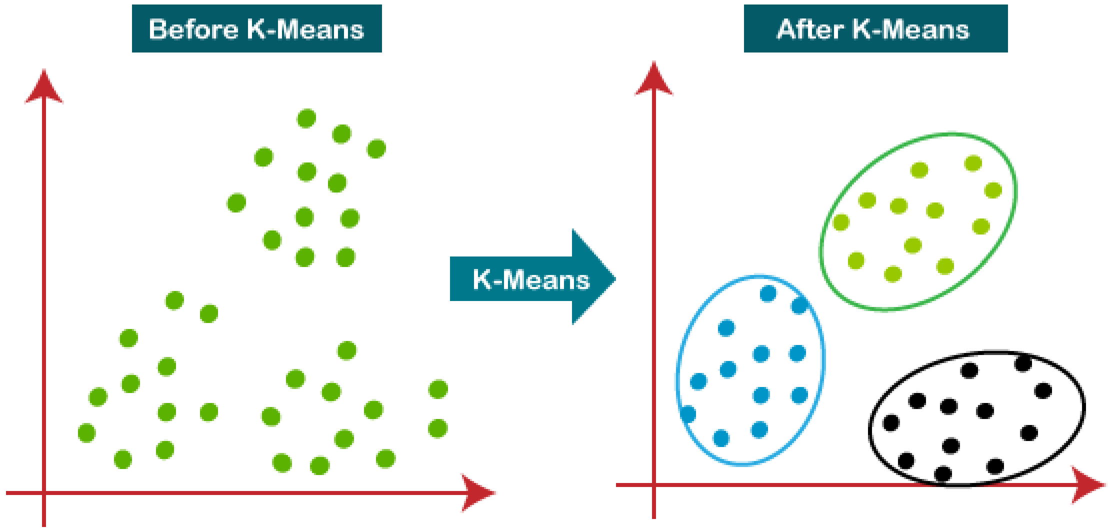


Figure 6. K Means Clustering analyzes unlabeled data and finds patterns amongst the values (Prishnu0, 2021).

Question: What Number of Clusters is best?

There are a number of methods which can be used to determine what number of clusters would work the best for a given problem (Matt.O, 2019). One method, aptly named the “Elbow Method”, looks for inflections in the Total Within Sum of Squares value. It is ideal that this number is minimized because it is a measure of variation within the cluster. Preferably, one would find the area on the graph where the slope quickly changes from steep to shallow (showing that the derivative is minimized so that the most information can be garnered with as few clusters as possible). This method is inexact – it is not looking for a specific statistically significant threshold, but it is based on the users observation of the inflection point. Not all inflection points may be obvious either. In Figure 7 below, there is a clear inflection point at k = 4 but there may be another one at k = 8. Since there were 8 classes in the dataset and there was a small inflection at k = 8, it was decided to make a second model with 8 clusters.

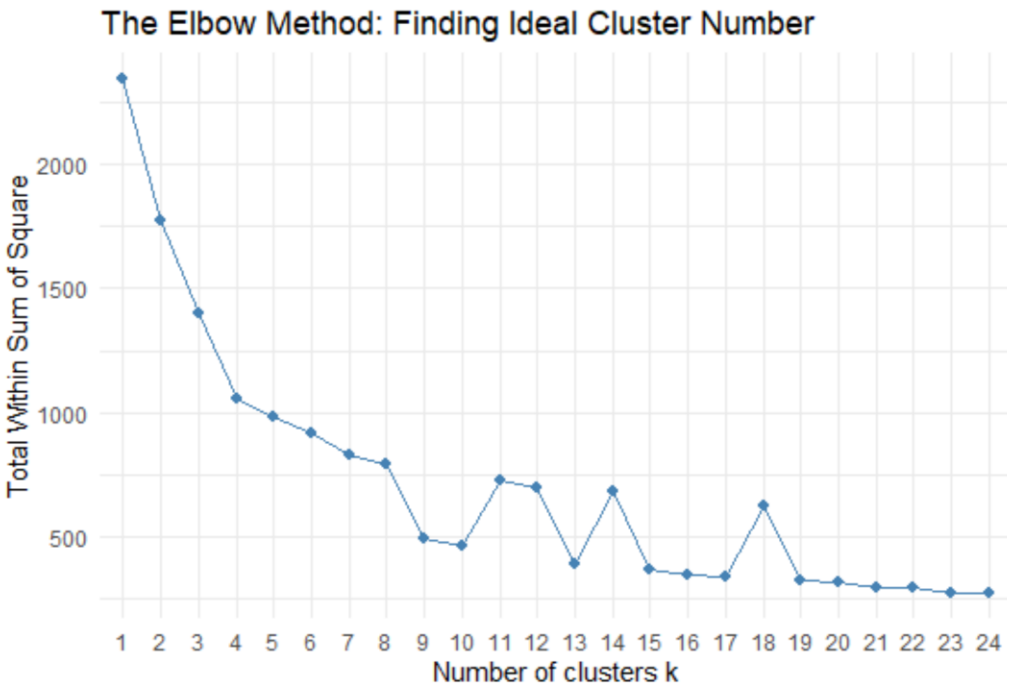


Figure 7. The Elbow Method shows an inflection point at k = 4 clusters and possibly another at k = 8 clusters.

Another method is named the Gap Statistic method. This is so named because it analyzes the “gap” between the variation under a null reference distribution of data versus the total within intracluster variation between different values of k (Matt.O, 2019). The graph below (Figure 8) automatically highlights when the gap statistic is maximized for as few clusters as possible. Again, k = 4 is shown to be the ideal number this time with statistical evidence to back it up.

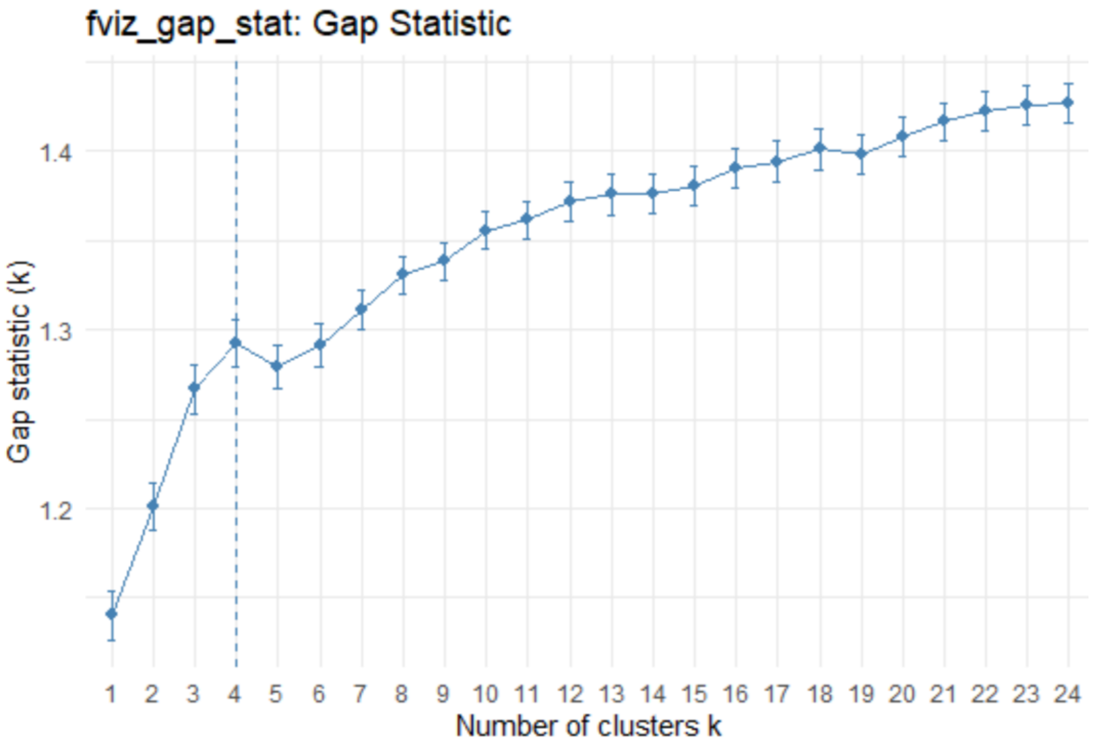
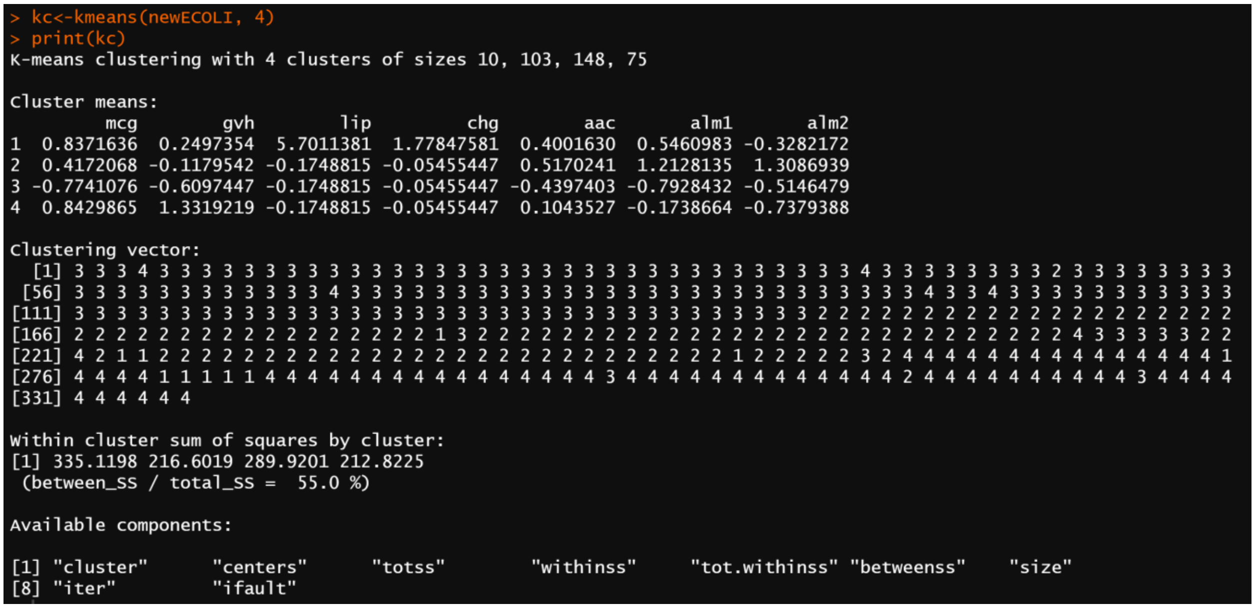


Figure 8. The Gap Statistic Method shows an inflection point and statistically determined the best cluster number is at k = 4.

Model 1:

The first model was created by simply following the recommended 4 clusters (based on Figures 7-8) using all data.

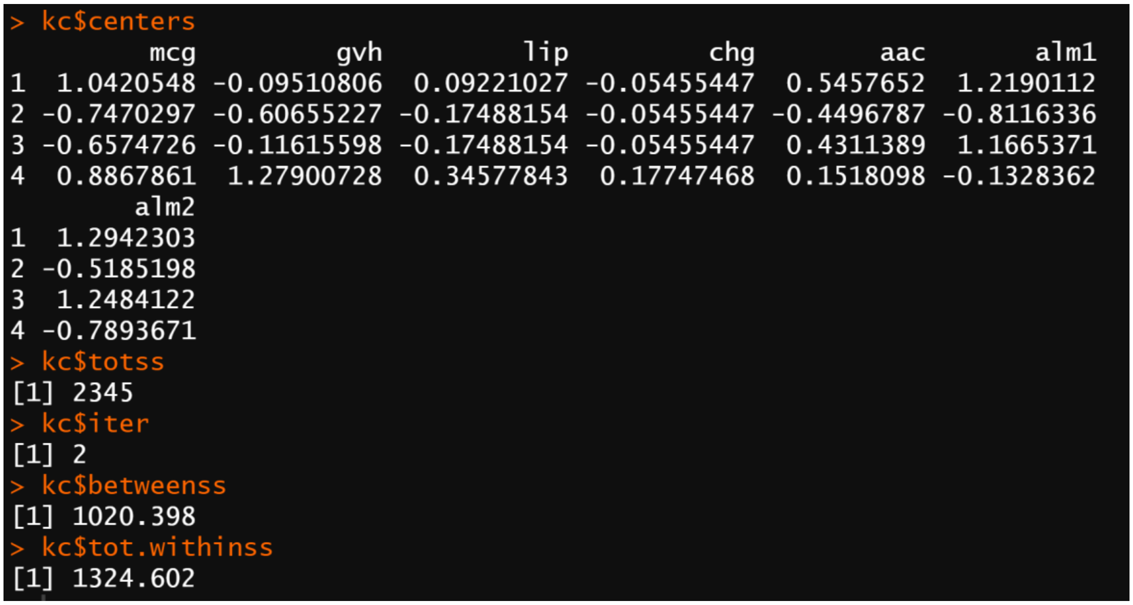


```
> kc<-kmeans(newECOLI, 4)
> print(kc)
K-means clustering with 4 clusters of sizes 10, 103, 148, 75

Cluster means:
        mcg        gvh        lip         chg        aac       alm1       alm2
1  0.8371636  0.2497354  5.7011381  1.77847581  0.4001630  0.5460983 -0.3282172
2  0.4172068 -0.1179542 -0.1748815 -0.05455447  0.5170241  1.2128135  1.3086939
3 -0.7741076 -0.6097447 -0.1748815 -0.05455447 -0.4397403 -0.7928432 -0.5146479
4  0.8429865  1.3319219 -0.1748815 -0.05455447  0.1043527 -0.1738664 -0.7379388

Clustering vector:
  [1] 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3
 [56] 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 4 3 3 3 3 3 3 3 3 3 3 3
[111] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[166] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 3 3 3 3 3 2 2
[221] 4 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 3 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1
[276] 4 4 4 4 1 1 1 1 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 3 4 4 4 4
[331] 4 4 4 4 4 4

Within cluster sum of squares by cluster:
[1] 335.1198 216.6019 289.9201 212.8225
 (between_SS / total_SS =  55.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
```

Figure 9. The output of the "kc" shows the summary of the method.

4 clusters were created of sizes: 10, 103, 148 and 75. The method shows the mean for each variable and for each cluster (note: this data is currently scaled in the image)(Figure 9). Next, the clustering vector shows all 336 rows of data in a vector and which cluster each row belongs to. The subsequent output of the kc method shows the within cluster sum of squares statistic. This measure shows how much variance exists within a cluster. It is critical to note that the larger the dataset is, the larger these values will be (they are relative to the dataset instead of absolute). Between_SS denotes the amount of variance between each cluster. Lastly, the Total_SS (total sum of squares) is a measure of the total variance within the dataset. Thus, altogether, the statistic of Bewteen_SS / Total_SS shows how much of the variance is explained by the clusters that were formed. As this number approaches 1 (or 100% as it is stylized in Figure 9), the model is better. The lower this number is, the worse the model is. Thus, the 55.0% seen in Figure 9 is not an excellent model since there is still 45.0% of the variance to explain which is not explained by the 4 clusters generated. The final row in the image shows the

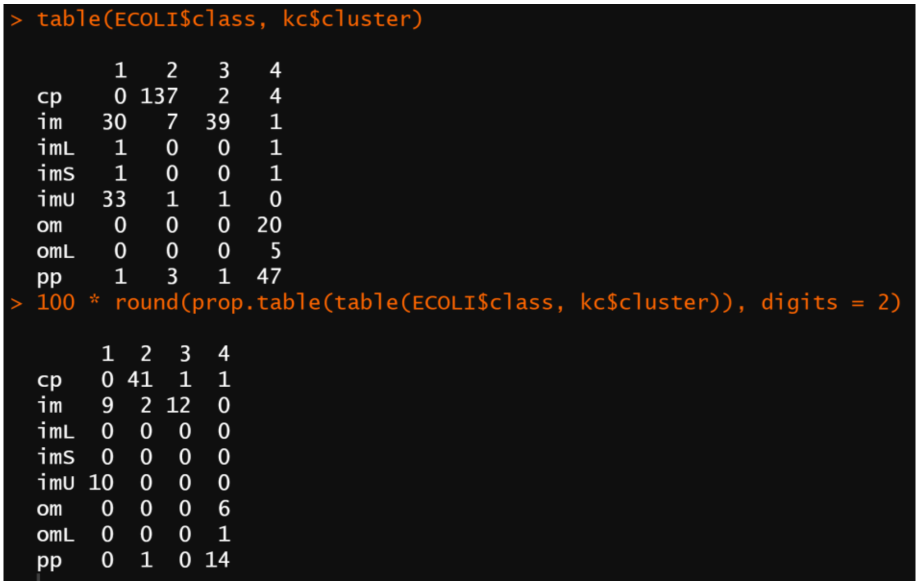
components that can be further called to see the statistics individually for the model (Figure 10). This shows some of the statistics already mentioned as well as the centers and number of iterations it took to create the model (written below as “iter”).



```
> kc$centers
         mcg         gvh         lip         chg        aac       alm1
1  1.0420548 -0.09510806  0.09221027 -0.05455447  0.5457652  1.2190112
2 -0.7470297 -0.60655227 -0.17488154 -0.05455447 -0.4496787 -0.8116336
3 -0.6574726 -0.11615598 -0.17488154 -0.05455447  0.4311389  1.1665371
4  0.8867861  1.27900728  0.34577843  0.17747468  0.1518098 -0.1328362
        alm2
1  1.2942303
2 -0.5185198
3  1.2484122
4 -0.7893671
> kc$totss
[1] 2345
> kc$iter
[1] 2
> kc$betweenss
[1] 1020.398
> kc$tot.withinss
[1] 1324.602
```

Figure 10. Calling the components of the kc method will show the individual statistics with which to evaluate each model.

Next, the distribution of classes amongst the clusters was found (both as the raw count and as the percentage of the total)(Figure 11). One might expect that with an optimal number of clusters, the data would parse nicely into the respective classes with a clear pattern. However, this was not the case for this data (because real-world data is often messy and not clear-cut). A majority of cluster 1 fell into inner membrane categories. The vast majority of cluster 2 was cytoplasmic protein (cp). Cluster 3 was mostly sequestered to inner membrane proteins (with a few others for cp, pp, and imU). Cluster 4 had the majority of values pertaining to the outer membrane and periplasm.



```
> table(ECOLI$class, kc$cluster)

       1   2   3   4
  cp   0 137   2   4
  im  30   7  39   1
  imL  1   0   0   1
  imS  1   0   0   1
  imU 33   1   1   0
  om   0   0   0  20
  omL  0   0   0   5
  pp   1   3   1  47
> 100 * round(prop.table(table(ECOLI$class, kc$cluster)), digits = 2)

      1  2  3  4
  cp  0 41  1  1
  im  9  2 12  0
  imL 0  0  0  0
  imS 0  0  0  0
  imU 10 0  0  0
  om  0  0  0  6
  omL 0  0  0  1
  pp  0  1  0 14
```

Figure 11. Table distribution between the clusters and the classes show no very clear pattern of dispersal.

A cluster plot of model 1 was made which showed visually there appeared to be a few outliers (Figure 12). This plot is based on a Principal Component Analysis (PCA) of the variables. PCA will reduce the number of variables while still capturing the information provided by those variables (Ding and He, 2004). Thus, the plot is able to show each instance plot across a 2D space. The line at the bottom of the plot essentially states that the PCA was able to contain 52.38% of the variability found within the data into the two components. In order to remedy this and improve the model, these outliers were identified.

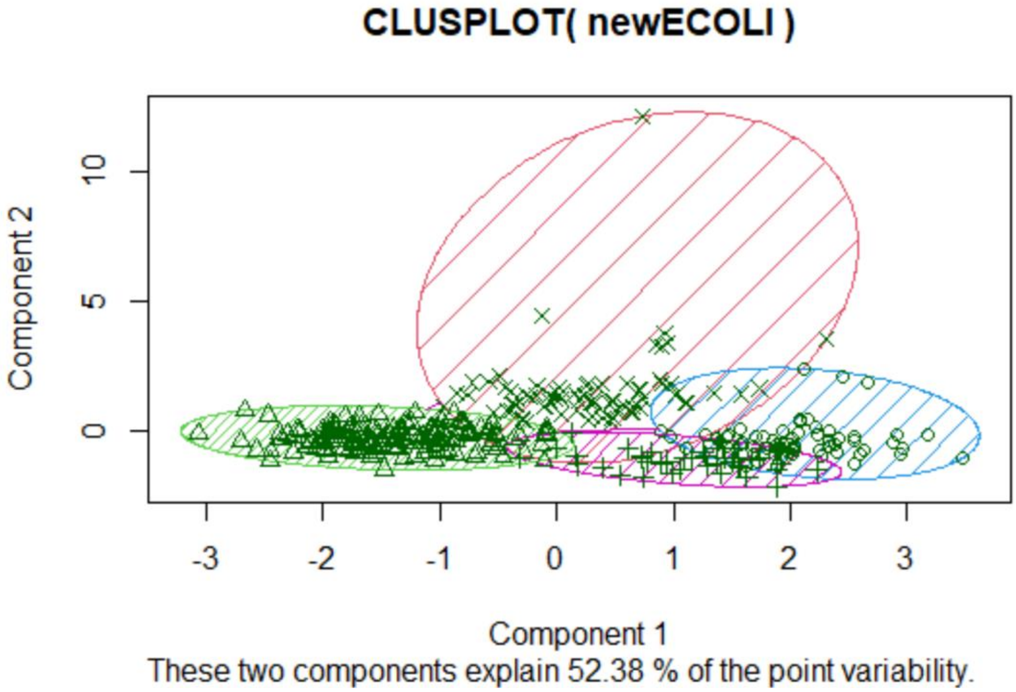
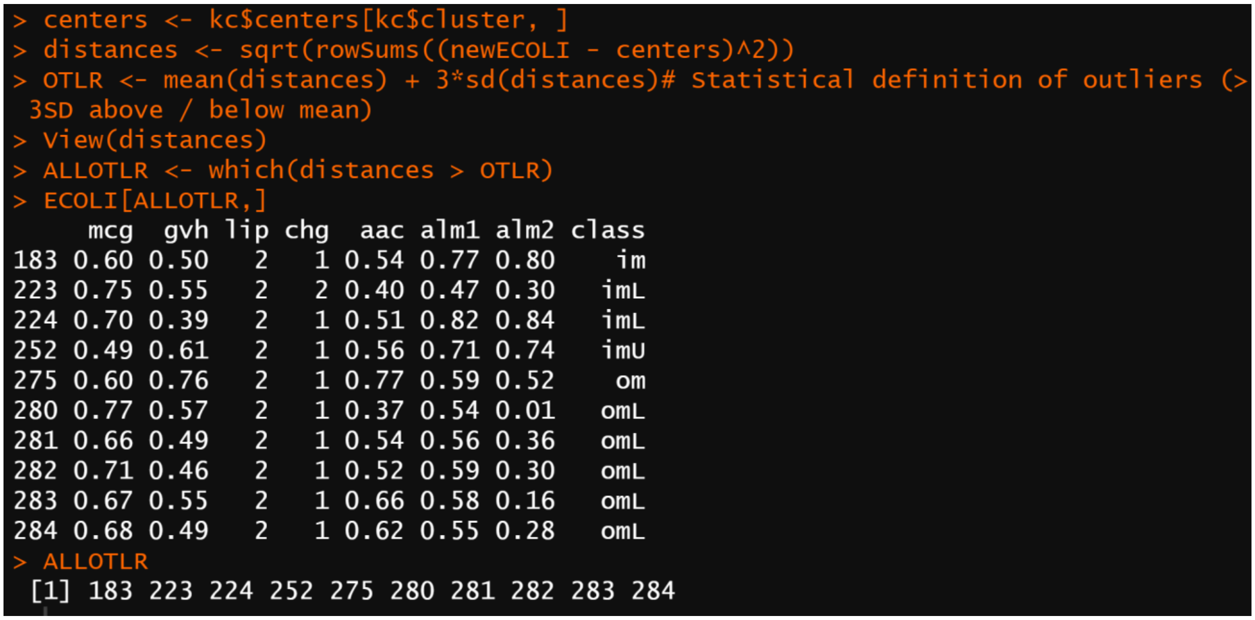


Figure 12. The plot of the 4 clusters shows there are a few apparent outliers with a majority of values existing closer together (1 = red, 2 = green, 3 = pink, and 4 = blue).



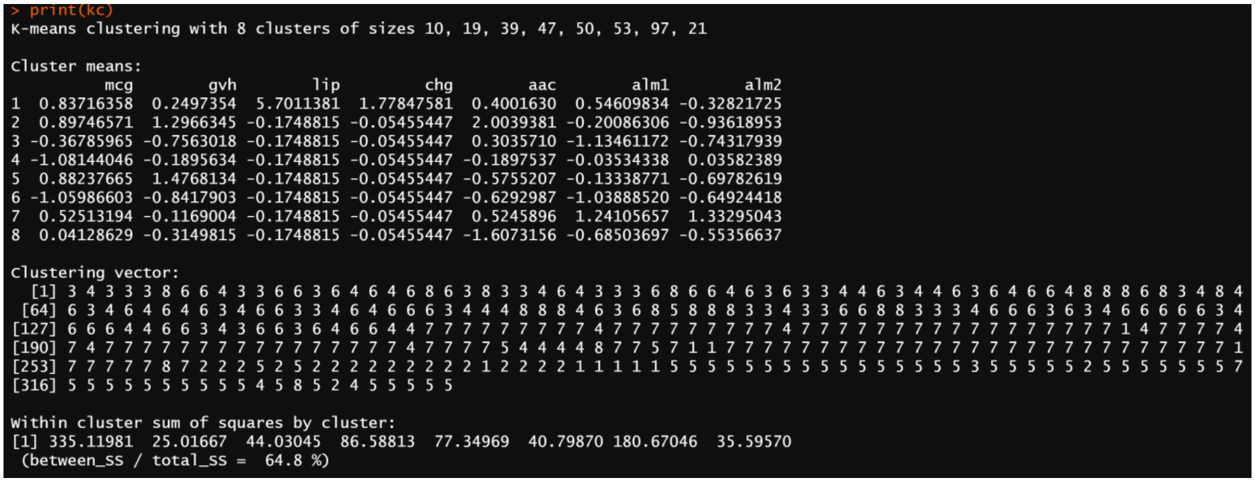
```
> centers <- kc$centers[kc$cluster, ]
> distances <- sqrt(rowSums((newECOLI - centers)^2))
> OTLR <- mean(distances) + 3*sd(distances)# Statistical definition of outliers (>
 3SD above / below mean)
> View(distances)
> ALLOTLR <- which(distances > OTLR)
> ECOLI[ALLOTLR,]
     mcg  gvh lip chg  aac alm1 alm2 class
183 0.60 0.50   2   1 0.54 0.77 0.80    im
223 0.75 0.55   2   2 0.40 0.47 0.30   imL
224 0.70 0.39   2   1 0.51 0.82 0.84   imL
252 0.49 0.61   2   1 0.56 0.71 0.74   imU
275 0.60 0.76   2   1 0.77 0.59 0.52    om
280 0.77 0.57   2   1 0.37 0.54 0.01   omL
281 0.66 0.49   2   1 0.54 0.56 0.36   omL
282 0.71 0.46   2   1 0.52 0.59 0.30   omL
283 0.67 0.55   2   1 0.66 0.58 0.16   omL
284 0.68 0.49   2   1 0.62 0.55 0.28   omL
> ALLOTLR
 [1] 183 223 224 252 275 280 281 282 283 284
```

Figure 13. Determining the values which had distances of greater than the mean plus 3 standard deviations highlighted 10 values.

Outliers were identified by first finding the centers. Then the distances were calculated for all of the values from their respective center. Outliers can be defined as being above or below 3 standard deviations (where if 1,000 samples are taken, only 3 will deviate from their expected range)(Altman and Bland, 2005). Thus, the standard deviation and mean of the distances were calculated. Those values which were greater than the mean (plus 3 standard deviations) were identified. There was an interesting trend observed: all 10 values identified had the binary value from the variable “lip” of 1 in the original dataset. One of the values (223) also was the only value in the whole dataset to contain the binary variable from the variable “chg” of 1 (this also had lip of 1). Since this seemed significant, it was decided a third model would be created with these values removed from the dataset.

Model 2:

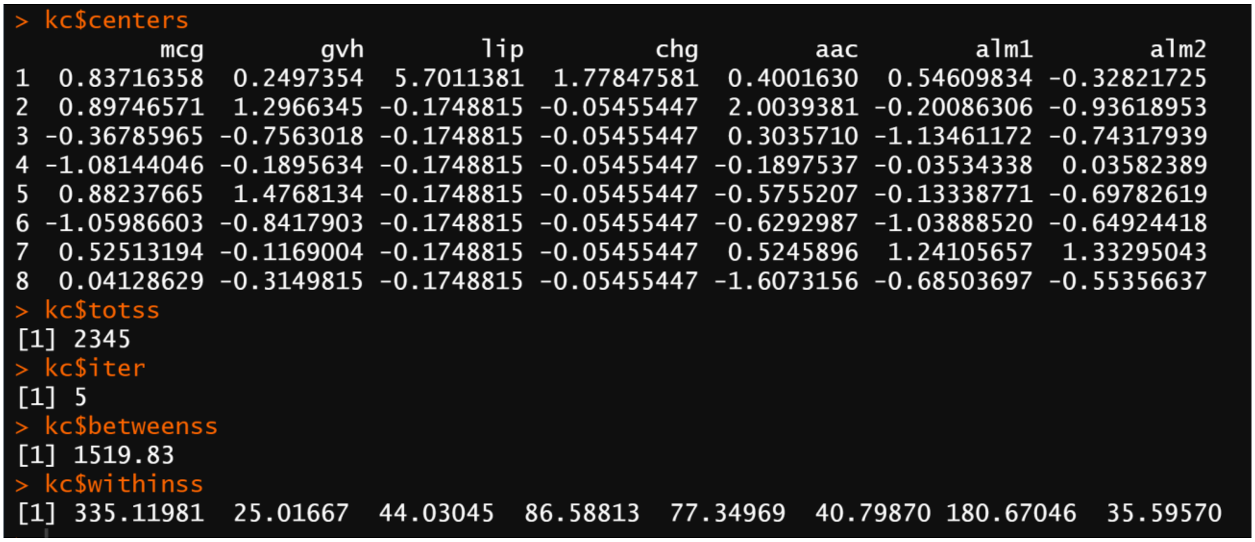
The second model was created by increasing the number to 8 clusters (based on Figures 7-8) using all data.



```
> print(kc)
K-means clustering with 8 clusters of sizes 10, 19, 39, 47, 50, 53, 97, 21

Cluster means:
          mcg        gvh        lip         chg        aac        alm1        alm2
1  0.83716358  0.2497354  5.7011381  1.77847581  0.4001630  0.54609834 -0.32821725
2  0.89746571  1.2966345 -0.1748815 -0.05455447  2.0039381 -0.20086306 -0.93618953
3 -0.36785965 -0.7563018 -0.1748815 -0.05455447  0.3035710 -1.13461172 -0.74317939
4 -1.08144046 -0.1895634 -0.1748815 -0.05455447 -0.1897537 -0.03534338  0.03582389
5  0.88237665  1.4768134 -0.1748815 -0.05455447 -0.5755207 -0.13338771 -0.69782619
6 -1.05986603 -0.8417903 -0.1748815 -0.05455447 -0.6292987 -1.03888520 -0.64924418
7  0.52513194 -0.1169004 -0.1748815 -0.05455447  0.5245896  1.24105657  1.33295043
8  0.04128629 -0.3149815 -0.1748815 -0.05455447 -1.6073156 -0.68503697 -0.55356637

Clustering vector:
  [1] 3 4 3 3 3 8 6 6 4 3 3 6 6 3 6 4 6 4 6 8 6 3 8 3 3 4 6 4 3 3 3 6 8 6 6 4 6 3 6 3 3 4 4 6 3 4 4 6 3 6 4 6 6 4 8 8 8 6 8 3 4 8 4
 [64] 6 3 4 6 4 6 4 6 3 4 6 6 3 3 4 6 4 6 6 6 3 4 4 4 8 8 8 4 6 3 6 8 5 8 8 8 3 3 4 3 3 6 6 8 8 3 3 3 4 6 6 6 3 6 3 4 6 6 6 6 6 3 4
[127] 6 6 6 4 4 6 6 3 4 3 6 6 3 6 4 6 6 4 4 7 7 7 7 7 7 7 7 7 4 7 7 7 7 7 7 7 7 7 4 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 1 4 7 7 7 7 4
[190] 7 4 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 4 7 7 7 7 5 4 4 4 4 8 7 7 5 7 1 1 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 1
[253] 7 7 7 7 7 8 7 2 2 2 5 2 5 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 1 1 1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 3 5 5 5 5 5 2 5 5 5 5 5 5 5 7
[316] 5 5 5 5 5 5 5 5 5 5 4 5 8 5 2 4 5 5 5 5 5

Within cluster sum of squares by cluster:
[1] 335.11981  25.01667  44.03045  86.58813  77.34969  40.79870 180.67046  35.59570
 (between_SS / total_SS =  64.8 %)
```

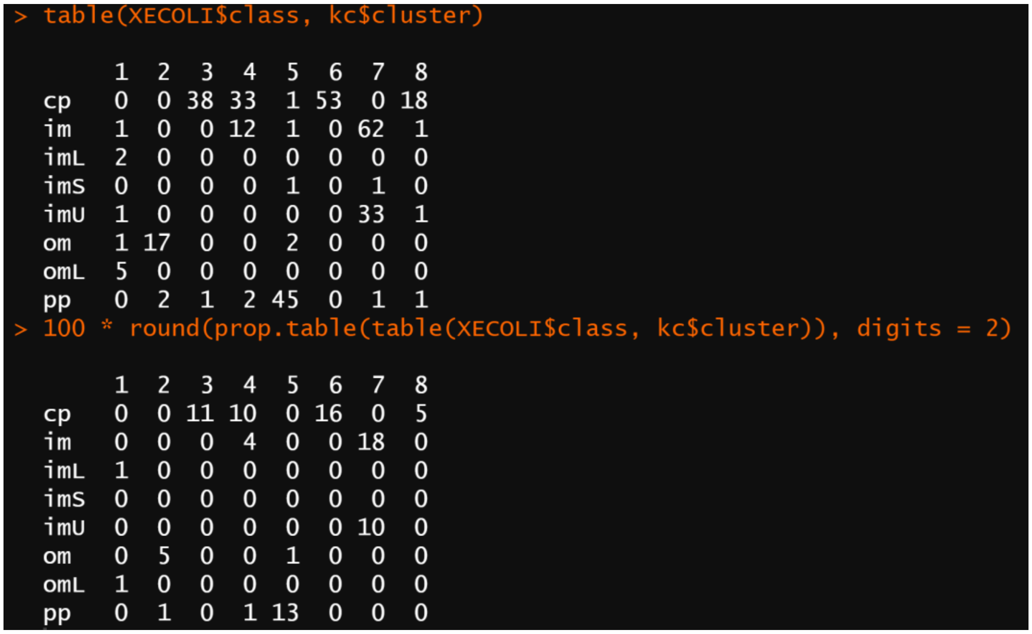
Figure 14. The output of the “kc” shows the summary of the method for model 2 with 8 clusters.

This model had a 64.8% between_SS / total_SS, a ~10% increase from model 1. One can also observe that the within sum of squares values had decreased but that is because these are relative values (to the dataset). Since there are less values in each cluster, the variance was also inherently lower.



```
> kc$centers
          mcg        gvh        lip         chg        aac        alm1        alm2
1  0.83716358  0.2497354  5.7011381  1.77847581  0.4001630  0.54609834 -0.32821725
2  0.89746571  1.2966345 -0.1748815 -0.05455447  2.0039381 -0.20086306 -0.93618953
3 -0.36785965 -0.7563018 -0.1748815 -0.05455447  0.3035710 -1.13461172 -0.74317939
4 -1.08144046 -0.1895634 -0.1748815 -0.05455447 -0.1897537 -0.03534338  0.03582389
5  0.88237665  1.4768134 -0.1748815 -0.05455447 -0.5755207 -0.13338771 -0.69782619
6 -1.05986603 -0.8417903 -0.1748815 -0.05455447 -0.6292987 -1.03888520 -0.64924418
7  0.52513194 -0.1169004 -0.1748815 -0.05455447  0.5245896  1.24105657  1.33295043
8  0.04128629 -0.3149815 -0.1748815 -0.05455447 -1.6073156 -0.68503697 -0.55356637
> kc$totss
[1] 2345
> kc$iter
[1] 5
> kc$betweenss
[1] 1519.83
> kc$withinss
[1] 335.11981  25.01667  44.03045  86.58813  77.34969  40.79870 180.67046  35.59570
```

Figure 15. Calling the components of the kc method will show the individual statistics with which to evaluate model 2 which was created using 5 iterations.



```
> table(XECOLI$class, kc$cluster)

       1   2   3   4   5   6   7   8
  cp   0   0  38  33   1  53   0  18
  im   1   0   0  12   1   0  62   1
  imL  2   0   0   0   0   0   0   0
  imS  0   0   0   0   1   0   1   0
  imU  1   0   0   0   0   0  33   1
  om   1  17   0   0   2   0   0   0
  omL  5   0   0   0   0   0   0   0
  pp   0   2   1   2  45   0   1   1
> 100 * round(prop.table(table(XECOLI$class, kc$cluster)), digits = 2)

       1   2   3   4   5   6   7   8
  cp   0   0  11  10   0  16   0   5
  im   0   0   0   4   0   0  18   0
  imL  1   0   0   0   0   0   0   0
  imS  0   0   0   0   0   0   0   0
  imU  0   0   0   0   0   0  10   0
  om   0   5   0   0   1   0   0   0
  omL  1   0   0   0   0   0   0   0
  pp   0   1   0   1  13   0   0   0
```

Figure 16. Table distribution between the clusters and the classes of model 2 show no clear pattern of dispersal.

The table distribution did not show a clear cut between each of the classes for the clusters. Instead, cytoplasmic proteins (cp) dominate across multiple clusters (3, 4, 6, & 8). Cluster 1 isn't dominated by any class. Cluster 2 is dominated by outer membrane protein. Cluster 5 is dominated by periplasmic protein. Cluster 7 is split between im and imU in a 2:1 ratio (Figure 16).

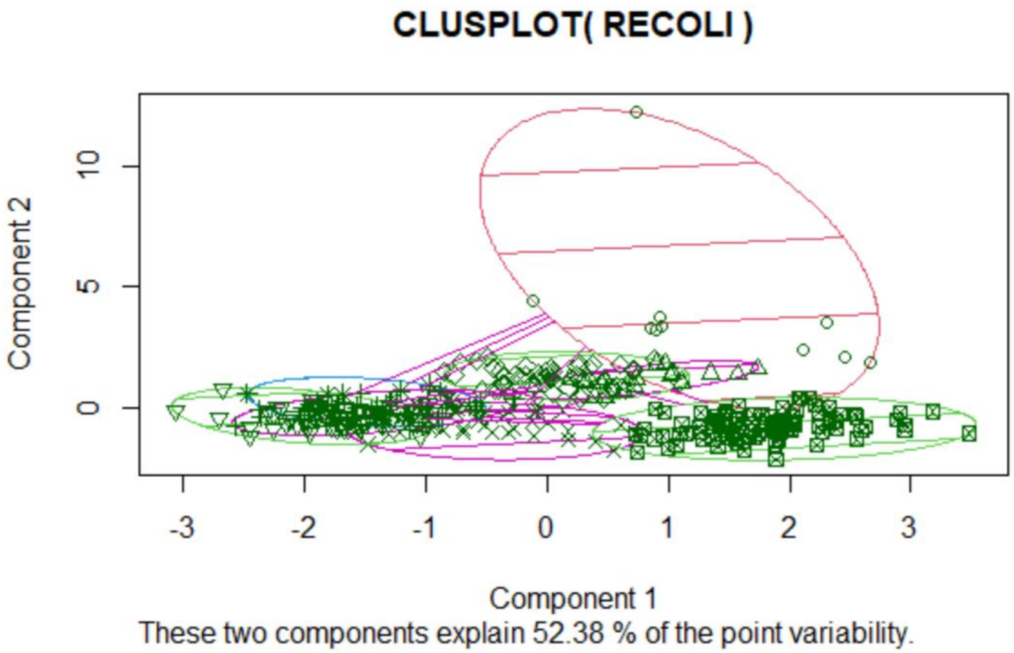
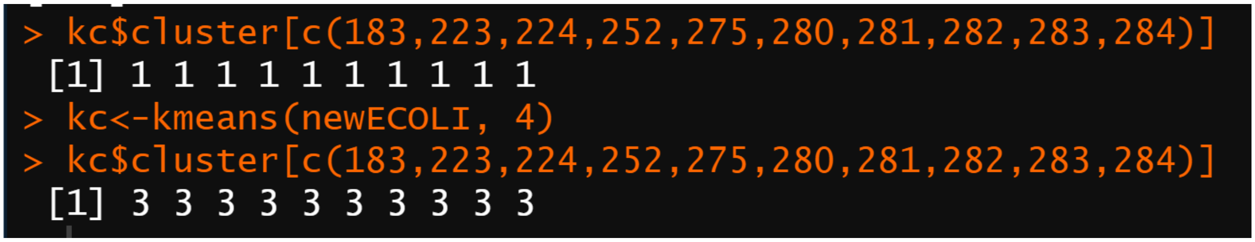


Figure 16. The plot of the 8 clusters shows the outliers are still dominating the analysis.

The cluster plot showed that the outliers were still dominating the analysis. The clusters appear to have primarily focused on circumscribing datapoints near where most of the dataset sits instead of going after the outliers. The PCA still only accounted for 52.38% of the variability within the data.

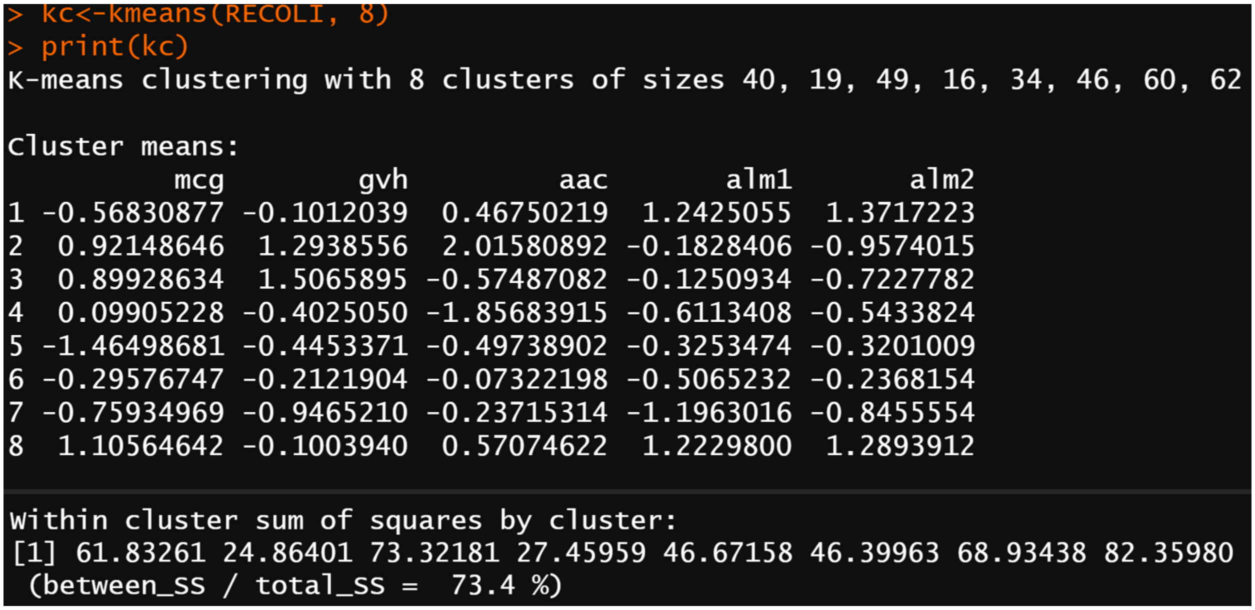
Model 3:

Finally, model 3 was created by removing the outliers from the dataset and then using 8 clusters. Since all the datapoints had a value of 1 with the variable of lip, both lip and chg were removed as variables. They were no longer of any added value since they would be homogenous across all other instances. The outlier values were also evaluated for what clusters they were lumped into for the previous models. In model 1, they were in cluster 3 and in model 2 they were in cluster 1 (Figure 17).



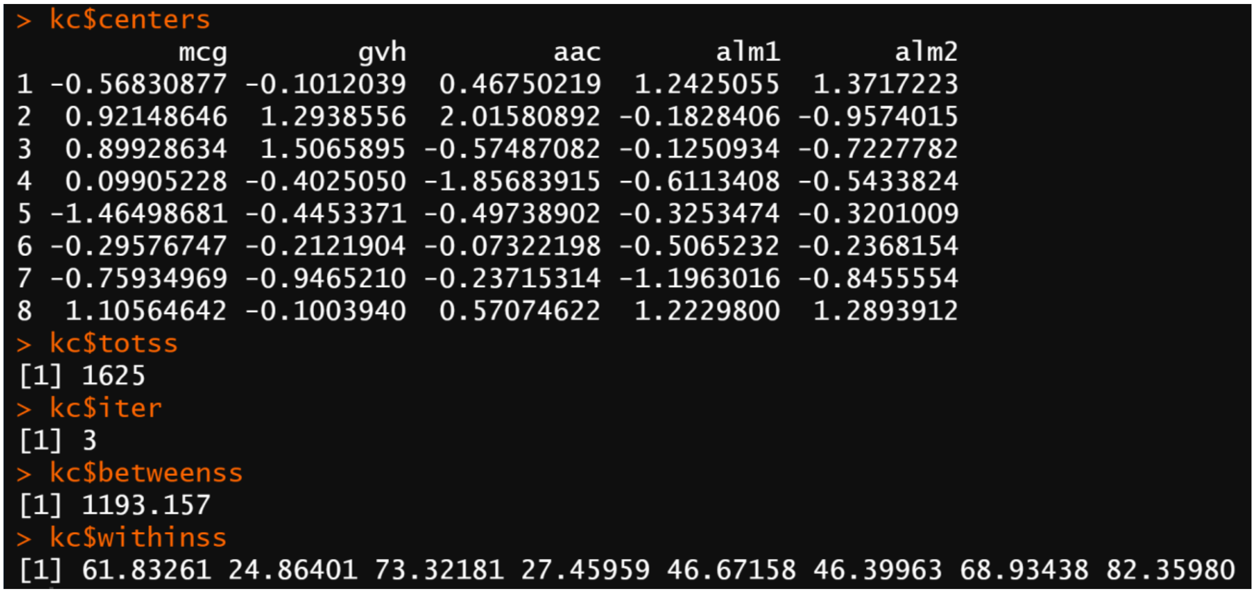
```
> kc$cluster[c(183,223,224,252,275,280,281,282,283,284)]
 [1] 1 1 1 1 1 1 1 1 1 1
> kc<-kmeans(newECOLI, 4)
> kc$cluster[c(183,223,224,252,275,280,281,282,283,284)]
 [1] 3 3 3 3 3 3 3 3 3 3
```

Figure 17. The 10 outlier values belonged to cluster 3 in model 1 and to cluster 1 in model 2.



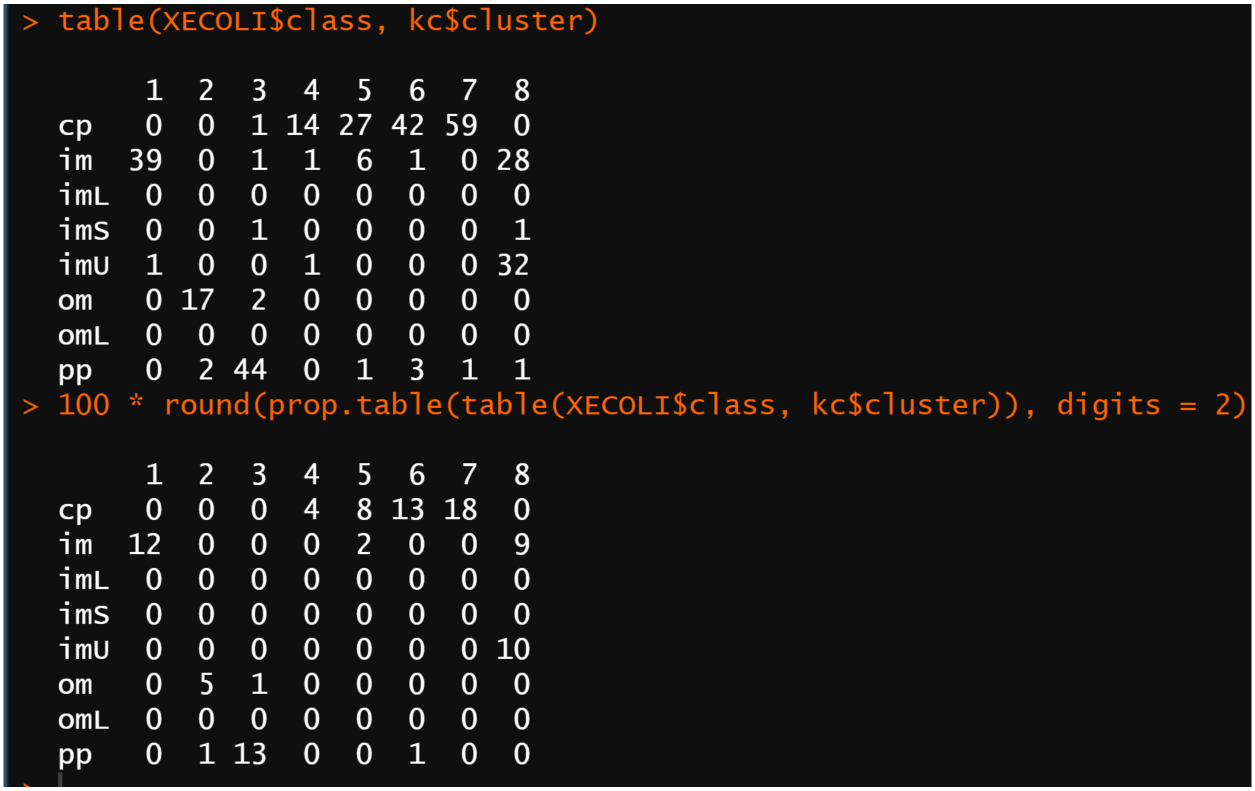
```
> kc<-kmeans(RECOLI, 8)
> print(kc)
K-means clustering with 8 clusters of sizes 40, 19, 49, 16, 34, 46, 60, 62

Cluster means:
          mcg        gvh         aac        alm1       alm2
1 -0.56830877 -0.1012039  0.46750219  1.2425055  1.3717223
2  0.92148646  1.2938556  2.01580892 -0.1828406 -0.9574015
3  0.89928634  1.5065895 -0.57487082 -0.1250934 -0.7227782
4  0.09905228 -0.4025050 -1.85683915 -0.6113408 -0.5433824
5 -1.46498681 -0.4453371 -0.49738902 -0.3253474 -0.3201009
6 -0.29576747 -0.2121904 -0.07322198 -0.5065232 -0.2368154
7 -0.75934969 -0.9465210 -0.23715314 -1.1963016 -0.8455554
8  1.10564642 -0.1003940  0.57074622  1.2229800  1.2893912

Within cluster sum of squares by cluster:
[1] 61.83261 24.86401 73.32181 27.45959 46.67158 46.39963 68.93438 82.35980
 (between_SS / total_SS =  73.4 %)
```

Figure 18. The output of the "kc" shows the summary of the method for model 3 with 8 clusters and no outliers.



```
> kc$centers
          mcg        gvh         aac       alm1       alm2
1 -0.56830877 -0.1012039  0.46750219  1.2425055  1.3717223
2  0.92148646  1.2938556  2.01580892 -0.1828406 -0.9574015
3  0.89928634  1.5065895 -0.57487082 -0.1250934 -0.7227782
4  0.09905228 -0.4025050 -1.85683915 -0.6113408 -0.5433824
5 -1.46498681 -0.4453371 -0.49738902 -0.3253474 -0.3201009
6 -0.29576747 -0.2121904 -0.07322198 -0.5065232 -0.2368154
7 -0.75934969 -0.9465210 -0.23715314 -1.1963016 -0.8455554
8  1.10564642 -0.1003940  0.57074622  1.2229800  1.2893912
> kc$totss
[1] 1625
> kc$iter
[1] 3
> kc$betweenss
[1] 1193.157
> kc$withinss
[1] 61.83261 24.86401 73.32181 27.45959 46.67158 46.39963 68.93438 82.35980
```

Figure 19. Calling the components of the kc method will show the individual statistics with which to evaluate model 3 which was created using 3 iterations.



```
> table(XECOLI$class, kc$cluster)

       1  2  3  4  5  6  7  8
  cp   0  0  1 14 27 42 59  0
  im  39  0  1  1  6  1  0 28
  imL  0  0  0  0  0  0  0  0
  imS  0  0  1  0  0  0  0  1
  imU  1  0  0  1  0  0  0 32
  om   0 17  2  0  0  0  0  0
  omL  0  0  0  0  0  0  0  0
  pp   0  2 44  0  1  3  1  1
> 100 * round(prop.table(table(XECOLI$class, kc$cluster)), digits = 2)

       1  2  3  4  5  6  7  8
  cp   0  0  0  4  8 13 18  0
  im  12  0  0  0  2  0  0  9
  imL  0  0  0  0  0  0  0  0
  imS  0  0  0  0  0  0  0  0
  imU  0  0  0  0  0  0  0 10
  om   0  5  1  0  0  0  0  0
  omL  0  0  0  0  0  0  0  0
  pp   0  1 13  0  0  1  0  0
```

Figure 20. Table distribution between the clusters and the classes of model 3 show no clear pattern of dispersal.

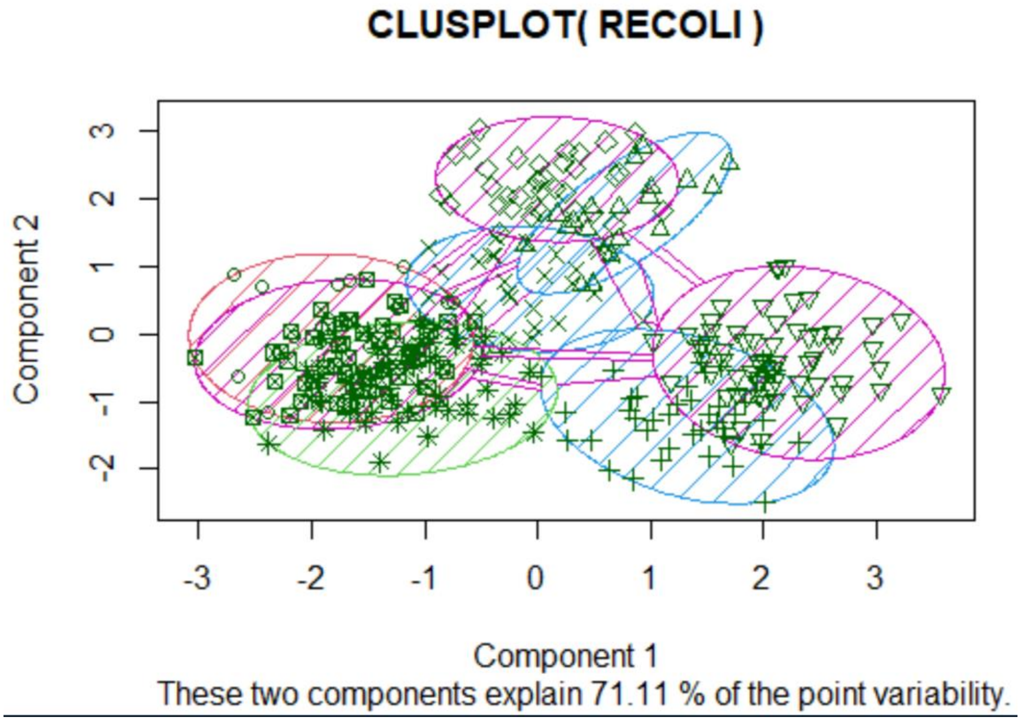


Figure 21. The plot of the 8 clusters with the outliers removed shows a more nuanced separation of the values.

Results and Model Evaluation:

Three models were made in total. The first model was made using 4 clusters per the recommendation of the elbow chart and the gap statistic (Figure 7 & 8). The second model was made using 8 clusters since there was a secondary inflection in the elbow chart (Figure 7) and because there were 8 classes. Lastly, the third model using 8 clusters and had the outliers removed. The evaluations of each model are seen below in Table 1.

	Clusters	**Between SS / Total SS**	**PCA**	**Comments**
Model 1	4 clusters	55.0%	52.38%	N/A
Model 2	8 clusters	64.8%	52.38%	N/A
Model 3	8 clusters	73.9%	71.11%	10 outliers removed

Table 1. Each model becomes about 10% more accurate from the former.

As a reminder, the PCA statistic is a measure of how much of the data variability is captured by the two components used to make the 2D plot of the clusters. So, this measures how well the model is able to able to summarize the data. This is why the PCA value is the same for the first two models – because the data is the same. But the 20% increase in the PCA value in the 3rd model is due to the 10 outliers being gone. Outliers by definition are harder to normalize and so removing them makes the data easier to reduce to the two components. This shows how drastically the output can change by removing outliers.

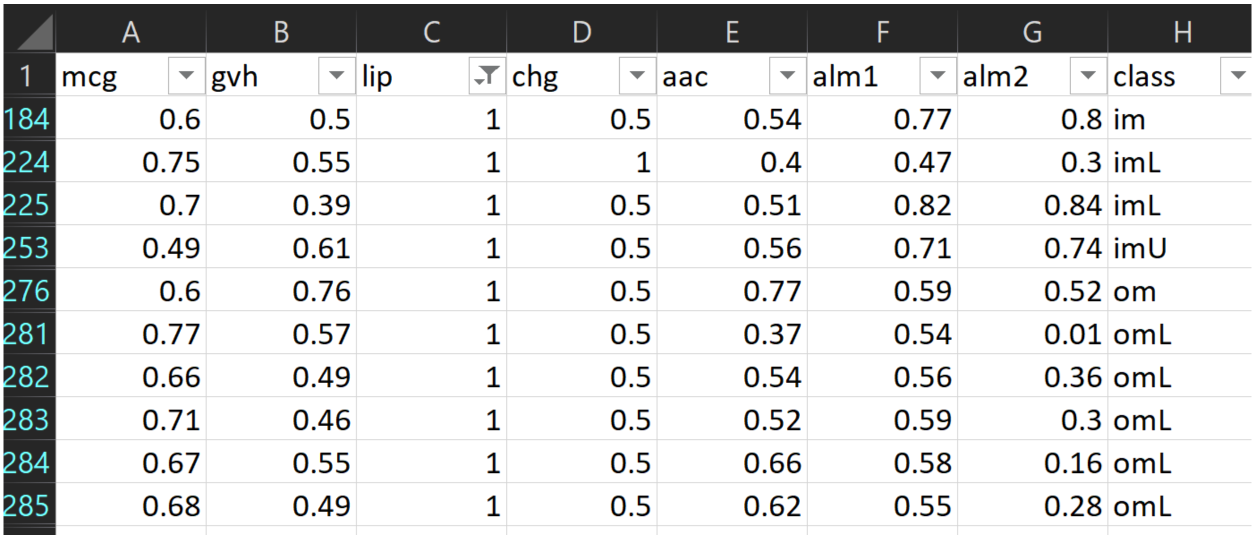
In contrast, the (Between_SS / Total_SS) statistic showed how well the model explains the variability found in the data. This value increased approximately 10% for each subsequent model. The ending value, 73.9%, denotes that over 70% of the variability seen in ECOLI is explained by the clusters created. In the context of ECOLI, this means that the clustering model can explain at least 70% of the variability in the data through 8 different buckets. However, when evaluating the buckets / clusters against the known classes, a few interesting patterns arise.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
cp	0	0	1	14	27	42	59	0
im	39	0	1	1	6	1	0	28
imL	0	0	0	0	0	0	0	0
imS	0	0	1	0	0	0	0	1
imU	1	0	0	1	0	0	0	32
om	0	17	2	0	0	0	0	0
omL	0	0	0	0	0	0	0	0
pp	0	2	44	0	1	3	1	1

Table 2. Analysis of the table of model 3 clusters versus the classes show imL and omL belong to no clusters.

Analysis of the model 3 clusters versus the classes show that each cluster has a single class that dominates it – these were highlighted in blue. However, curiously not all classes were represented in the output. Cytoplasmic (cp) dominated in 4 clusters: 4, 5, 6, and 7. This could indicate this model is detecting multiple different types of proteins within the cp class. Since this category held the most values in general (143/336), it is likely that there would be variation within the class. Furthermore, the cytoplasm in a cell is by far the largest area in a cell. Because of this, it is not only surprising that cytoplasmic proteins represent a massive amount of this dataset, but also that they would have different types within the class of “cytoplasmic”. Clusters 1, 2, and 3 were dominated by inner membrane proteins, outer membrane proteins, and periplasmic proteins respectively.

imU (or inner membrane with an uncleavable signal sequence) did not dominate in any category. It only had 2 values in the first place as the smallest category. These values were split between clusters 4 and 8 and could not have dominated. Lastly, the classes omL and imL both had no values in them and could not have dominated in any cluster. These originally had 5 and 2 values in them respectively, but these were removed as part of the outlier removal (Figure 22). The last of the 3 outliers were from the classes im, imU, and om. Removing 1 value from each of these classes did not significantly lower the sum of those classes.



	A	B	C	D	E	F	G	H
1	mcg	gvh	lip	chg	aac	alm1	alm2	class
184	0.6	0.5	1	0.5	0.54	0.77	0.8	im
224	0.75	0.55	1	1	0.4	0.47	0.3	imL
225	0.7	0.39	1	0.5	0.51	0.82	0.84	imL
253	0.49	0.61	1	0.5	0.56	0.71	0.74	imU
276	0.6	0.76	1	0.5	0.77	0.59	0.52	om
281	0.77	0.57	1	0.5	0.37	0.54	0.01	omL
282	0.66	0.49	1	0.5	0.54	0.56	0.36	omL
283	0.71	0.46	1	0.5	0.52	0.59	0.3	omL
284	0.67	0.55	1	0.5	0.66	0.58	0.16	omL
285	0.68	0.49	1	0.5	0.62	0.55	0.28	omL

Figure 22. The 10 outliers were all of the values in the classes imL and omL.

It was surprising that 8 clusters performed better than 4 clusters since 4 was the recommended value to be used by two different methods. It does beg the question if more clusters would always reduce the between_SS / total_SS value. It logically follows that the more clusters there are, the more the variability in the data would be explained by those clusters (up until the point where each datapoint has its own cluster). However, too many clusters is pointless which is why there are methods like the elbow method and gap statistic to determine the ideal number of clusters.

<u>Conclusion:</u>

Proteins are the reason for life. It is critical to understand them in order to understand anything else about cellular biology. Protein localization prediction is one of the basic elements of cellular biology that allows scientists to make other advances. There are other methods that can work in tandem with computational methods to help confirm the results of the methods. For example, Green Fluorescent Protein (GFP) can be attached genetically to a protein so that wherever the protein localizes, it can be visually observed (Feilmeier et al., 2000). This requires the use of a fluorescence microscope. It also requires using a plasmid (or another DNA splicing method like CRSPR-Cas9). It can also be used to detect protein folding which will affect protein localization as well (Feilmeier et al., 2000).

As previously mentioned, it is a known fact that protein sequence and particular structures can be used to predict a protein's localization site (Sjöström et al., 1987). Their end destination in particular can be detected by their signal peptides, since this is what is generally used by transporters in the cell. These transporters "read" the peptide sequence (which is a small region of the protein) and use it as an address of where to take that protein. With this in mind, it was shown in this study that protein localization sites could be explained with >70% accuracy. This was done by finding similarities in the data. If the dataset did not contain the class variable, the method used could have found patterns in the data. It would have highlighted that several clusters mostly belonged to a single localization site. This type of program would be particularly helpful if one had a dataset of raw protein values but needed to know which localization site each protein likely belonged to.

Limitations and Improvements:

There are multiple areas this could be improved and ways in which this study was limited. First, the dataset needed to have more equal classes. Because the 8 classes of protein were so uneven, one could not accurately assess them evenly. Even if relatively equal numbers cannot be garner, then classes should only be analyzed if they are above a certain threshold (like 20). Adjacently to this, the dataset needed to have more values in the binary categories. All of the values in the minority of the two variables lip and chg were detected as outliers. This is likely because this was the starkest difference detected amongst the data. If the outlier detection algorithm was run again after those 10 values were removed, it is possible more datapoints would have been detected as outliers as well. Next, it would be fascinating to look into why model 3 put the cytoplasmic proteins into 4 different clusters. It would be interesting to look at what those proteins were and detect if there were critical similarities between them in their structure. If not, then one could look back at the data to find out why they were grouped together as one. Finally, it would be fascinating to compare these results to previously documented results of E. coli studies to detect the differences between cp proteins and answer some of the other questions discussed here.

References:

Altman, D. G., & Bland, J. M. (2005). Standard deviations and standard errors. *Bmj*, *331*(7521), 903.

Berg, H. C. (2008). *E. coli in Motion*. Springer Science & Business Media.

Ding, C., & He, X. (2004). K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning* (p. 29).

Feilmeier, B. J., Iseminger, G., Schroeder, D., Webber, H., & Phillips, G. J. (2000). Green fluorescent protein functions as a reporter for protein localization in Escherichia coli. *Journal of bacteriology*, *182*(14), 4068-4076.

Han, J., Kamber, M., and Pei, J. (2011). Data Mining: Concepts and Techniques, Third Edition. Elsevier. Retrieved June 6th, 2021 from:

http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

Horton, P. & Nakai, K. (1996). "A Probablistic Classification System for Predicting the Cellular Localization Sites of Proteins". Intelligent Systems in Molecular Biology.

Kaeberlein, M., & Kennedy, B. K. (2007). Protein translation, 2007. *Aging cell*, *6*(6), 731-734.

Matt.O. (2019). “10 Tips for Choosing the Optimal Number of Clusters”. Towards Data Science. https://towardsdatascience.com/10-tips-for-choosing-the-optimal-number-of-clusters-277e93d72d92

MBInfo. (n.d.). Protein Localization. MechanoBio. https://www.mechanobio.info/the-cell/protein-localization/

Pranshu0. (2021). K Means Clustering Simplified in Python. Analytics Vidhya. Retrieved from: https://www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python/

Sjöström, M., Wold, S., Wieslander, A., & Rilfors, L. (1987). Signal peptide amino acid sequences in Escherichia coli contain information related to final protein localization. A multivariate data analysis. *The EMBO journal*, *6*(3), 823-831.

UCI Machine Learning Lab (1996). Ecoli Data Set. UCI. Retrieved from: http://archive.ics.uci.edu/ml/datasets/Ecoli

von Heijne G. (1986). A new method for predicting signal sequence cleavage sites. Nucleic acids research, 14(11), 4683–4690. https://doi.org/10.1093/nar/14.11.4683

Weber, A. L., & Miller, S. L. (1981). Reasons for the occurrence of the twenty coded protein amino acids. *Journal of Molecular Evolution*, *17*(5), 273-284.