Lab 5: CLustering

Jeff McGovern and Nicole Martin
CSC 466, Section 03
December 5, 2015

Contents

1 Introduction

Two different unsupervised learning techniques were implemented for this lab: k-means clustering and agglomerative clustering.

This assignment parsed CSV and TXT files that represented the actual data and the header specifying data attribute names respectively. The supplied data sets contained both real data sets and synthetic data sets, as described below.

- 4clusters - synthetic data set of 4 clusters in 2D space. Data consists of 2 columns representing x and y coordinates of points in 2D space.
- mammal_milk - animals and the percentage of different constituents in their milk. Data consists of the name of the mammal and the percentage of constituents such as water, protein, fat, lactose, and ash.
- economy - profit vs equity in sectors of the economy. Profit as a percentage of stockholder's equity and data is per the Reader's Digest Almanac of 1966
- planets - sightings of minor planets. Data consists of the year and intitals of the astronomer, the angle (in the earth's plane of orbit) at which the minor planet crossed the earth's orbit (Node), the angle between the orbits of the earth and the minor planet (Inclination), and the maximum distance between the minor planet and the sun divided by the corresponding quantity for the earth (Axis).
- iris - measurements of different Iris flowers. Data consists of sepal length in cm, sepal width in cm, petal length in cm, petal width in cm, and the Iris class (Iris Setosa, Iris Versicolour, or Iris Virginica).
- many_clusters - synthetic data set of many clusters in 2D space. Data consists of 2 columns representing x and y coordinates of points in 2D space.
- AccidentsSet01 - fatal automotive accidents. The number of vehicles, people, and fatalities are provided for each accident.
- AccidentsSet02 - fatal automotive accidents. For each accident, the number of vehicles, number of people, number of lanes, speed limit, number of fatalities, and number of drunk drivers involved are provided.
- AccidentsSet03 - fatal automotive accidents. For each accident, the number of vehicles, number of pedestrians, number of lanes, number of fatalities, and number of drunk drivers are provided.

2 Study Design

2.1 Language

Two clustering methods were implemented: k-means clustering and agglomerative hierarchical clustering. Both clustering methods were implemented in Python 3.5 on an Intel i7 4770k processor and an Intel i3-5005U processor. Notable packages common to both include `argparse` for parsing complicated command line arguments. Clusters are represented as Python lists in order to allow for duplicates to appear in the cluster, since the closest implementation of a multiset simply counts the objects instead of duplicating them.

Including the optional header file via `[Header_Filename]` is only supported for header files that are single-line descriptions of the CSV (e.g. `VE_TOTAL, PERSONS, FATALS`). Certain header files are simply a freeform description of the data and are not supported. If the `-i` flag is included, the program will attempt to find the appropriate header file based on the supplied CSV.

2.2 *k*-means clustering

The help text for *k*-means clustering is below:

```
./kmeans.py -h
usage: ./kmeans.py [-h] [-i] [-e | -t | -d | -p]
                   <Filename> <k> [Header_Filename]

k-Means Clustering

positional arguments:
  <Filename>          name of the CSV file containing the input dataset
  <k>                 number of clusters the program has to produce
  Header_Filename     name of the CSV file containing the input dataset's header

                   If [Header_Filename] IS NOT provided, the program can deduce the header
                   file from the data filename by prepending 'header_' to it, using a
                   '.txt' extension instead, and looking in the working directory,
                   by adding the [-i | --infer-header] flag to the arguments.

optional arguments:
  -h, --help          show this help message and exit
  -i, --infer-header  infer the header from the data file filename
  -e, --euclidean     distance metric: euclidean (default)
  -t, --taxicab       distance metric: taxicab (manhattan)
  -d, --dot           distance metric: dot product
  -p, --pearson       distance metric: pearson correlation
```

The `-e`, `-t`, `-d`, and `-p` flags refer to the distance metric used by *k*-means and stand for euclidean, taxicab (manhattan), dot product, and pearson correlation respectively. If no distance metric is specified, euclidean distance is used by default.

The *k*-means algorithm implemented is Bing Liu's disk *k*-means algorithm. The selected implementation of *k*-means clustering selected the initial cluster centroids randomly. Figure A.1 shows the Python implementation of the algorthm.

2.3 Agglomerative clustering

Below is the help output for the hierarchical clustering program.

```
./hclustering.py -h
usage: ./hclustering.py [-h] [-i] [-s | -c | -a | -m] [-e | -t | -d | -p]
                        <Filename> [threshold] [Header_Filename]
```

```
Hierarchical Clustering

positional arguments:
  <Filename>          name of the CSV file containing the input dataset
  threshold           optional threshold at which the program will "cut" the
                      cluster hierarchy to report the clusters:

                   If <threshold> parameter IS specified in the input, the
                   program shall produce both the cluster hierarchy, and the
                   appropriate list of clusters cut at the specified
                   threshold.

                   If <threshold> parameter IS NOT specified in the input, the
                   program shall produce the cluster hierarchy alone.
  Header_Filename     name of the CSV file containing the input dataset's header

                   If [Header_Filename] IS NOT provided, the program can deduce the header
                   file from the data filename by prepending 'header_' to it, using a
                   '.txt' extension instead, and looking in the working directory,
                   by adding the [-i | --infer-header] flag to the arguments.

optional arguments:
  -h, --help          show this help message and exit
  -i, --infer-header  infer the header from the data file filename
  -s, --single        cluster distance: single link (default)
  -c, --complete      cluster distance: complete link
  -a, --average       cluster distance: average link
  -m, --centroid      cluster distance: centroid method
  -e, --euclidean     distance metric: euclidean (default)
  -t, --taxicab       distance metric: taxicab (manhattan)
  -d, --dot           distance metric: dot product
  -p, --pearson       distance metric: pearson correlation
```

The hierarchical clustering method used is agglomerative clustering, outlined in the pseudocode below. Distance metrics are the distance calculation between datapoints, while cluster distances are the strategy of distance computation between clusters used in agglomeration. Euclidean distance remained the default distance metric while single link was the default cluster distance.

The dendrograms of the clustered datasets are output in XML format. Notably, creating the XML representation relied heavily on the `xml.etree.ElementTree` package in Python, allowing for efficient, recursive, and easy to read building of the XML tree. In order to cut off at a threshold, we employed the strategy in Figure A.1:

Algorithm 1 Bing Liu's Agglomerative Clustering Algorithm

procedure AGGLOMERATIVE(D)
  Make each datapoint in the data set D a cluster,
  Compute all pairwise distances of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in D$;
  repeat
    find two clusters that are nearest to each other;
    merge the two clusters to form a new cluster c;
    compute the distance from c to all other clusters;
  until there is only one cluster left
end procedure

3 Results and Discussion

3.1 4clusters

3.1.0.1 Best k-means clusters Given that the synthetic data set 4clusters.csv was designed to have for clusters, $k = 4$ was selected. Due to the randomized cluster centers, it took several attempts to get a k-means clustering which overall minimized the euclidean distance to center and sum squared errors for all the clusters. The best clusters from using $k = 4$ and are shown in Tables 1, 2, 3, and 4.

Table 1: 4clusters Cluster 0 data from k-means clustering, euclidean distance $k = 4$

Center	:
33.166666666666664	17.27777777777778
Size	: 18
Min Dist to Center	: 2.283867
Max Dist to Center	: 12.124101
Avg Dist to Center	: 7.579450
Sum Squared Error	: 136.430103
32.0	27.0
26.0	25.0
39.0	24.0
34.0	23.0
37.0	23.0
22.0	22.0
38.0	21.0
35.0	20.0
31.0	18.0
26.0	16.0
31.0	13.0
26.0	16.0
38.0	13.0
29.0	11.0
34.0	11.0
37.0	10.0
40.0	9.0

Table 1: 4clusters Cluster 0 data from k-means clustering, euclidean distance $k = 4$

Center	:
33.166666666666664	17.27777777777778
Size	: 18
Min Dist to Center	: 2.283867
Max Dist to Center	: 12.124101
Avg Dist to Center	: 7.579450
Sum Squared Error	: 136.430103
42.0	9.0

Table 2: 4clusters Cluster 1 data from k-means clustering, euclidean distance $k = 4$

Center	:
22.0	38.0
Size	: 2
Min Dist to Center	: 3.000000
Max Dist to Center	: 3.000000
Avg Dist to Center	: 3.000000
Sum Squared Error	: 6.000000
19.0	38.0
25.0	38.0

Table 3: 4clusters Cluster 2 data from k-means clustering, euclidean distance $k = 4$

Center	:
9.9	37.8
Size	: 10
Min Dist to Center	: 0.921954
Max Dist to Center	: 4.341659
Avg Dist to Center	: 3.429930
Sum Squared Error	: 34.299301
10.0	42.0
8.0	41.0
13.0	40.0
7.0	39.0
9.0	38.0
12.0	38.0
6.0	37.0
13.0	35.0
9.0	34.0
12.0	34.0

Table 4: 4clusters Cluster 3 data from k-means clustering, euclidean distance $k = 4$

Center	:
41.111111111111114	41.77777777777778
Size	: 9
Min Dist to Center	: 0.785674
Max Dist to Center	: 4.275974
Avg Dist to Center	: 2.911701
Sum Squared Error	: 26.205306
41.0	45.0
39.0	44.0
42.0	43.0
44.0	43.0
38.0	42.0
41.0	41.0
45.0	40.0
38.0	39.0
42.0	39.0

3.1.0.2 Best Agglomerative clusters

Agglomerative clustering was initially run with euclidean distance, single link cluster distance, and no threshold to view the full tree structure and determine a good threshold for assigning clusters. It was found that using the centroid method with euclidean distance and a threshold of 12.00 produced the clusters with the best fit. Clusters are described in Tables 5, 6, 7, and 8.

Table 5: 4clusters Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 12.00

Center	:
9.9	37.8
Size	: 10
Min Dist to Center	: 0.921954
Max Dist to Center	: 4.341659
Avg Dist to Center	: 3.429930
Sum Squared Error	: 34.299301
9.0	34.0
13.0	35.0
12.0	34.0
13.0	40.0
12.0	38.0
10.0	42.0
8.0	41.0
6.0	37.0
7.0	39.0
9.0	38.0

Table 6: 4clusters Cluster 1 data from agglomerative clustering, euclidean distance centroid method threshold = 12.00

Center	:
22.0	38.0
Size	: 2
Min Dist to Center	: 3.000000
Max Dist to Center	: 3.000000
Avg Dist to Center	: 3.000000
Sum Squared Error	: 6.000000
19.0	38.0
25.0	38.0

Table 7: 4clusters Cluster 2 data from agglomerative clustering, euclidean distance centroid method threshold = 12.00

Center	:
41.111111111111114	41.77777777777778
Size	: 9
Min Dist to Center	: 0.785674
Max Dist to Center	: 4.275974
Avg Dist to Center	: 2.911701
Sum Squared Error	: 26.205306
38.0	42.0
38.0	39.0
42.0	43.0
44.0	43.0
41.0	45.0
39.0	44.0
45.0	40.0
41.0	41.0
42.0	39.0

Table 8: 4clusters Cluster 3 data from agglomerative clustering, euclidean distance centroid method threshold = 12.00

Center	:
33.166666666666664	17.27777777777778
Size	: 18
Min Dist to Center	: 2.283867
Max Dist to Center	: 12.124101
Avg Dist to Center	: 7.579450
Sum Squared Error	: 136.430103
40.0	9.0
42.0	9.0
34.0	11.0
38.0	13.0

Table 8: 4clusters Cluster 3 data from agglomerative clustering, euclidean distance centroid method threshold = 12.00

Center	:
33.166666666666664	17.27777777777778
Size	: 18
Min Dist to Center	: 2.283867
Max Dist to Center	: 12.124101
Avg Dist to Center	: 7.579450
Sum Squared Error	: 136.430103
37.0	10.0
32.0	27.0
38.0	21.0
39.0	24.0
37.0	23.0
34.0	23.0
35.0	20.0
26.0	25.0
22.0	22.0
31.0	13.0
29.0	11.0
31.0	18.0
26.0	16.0
26.0	16.0

3.1.0.3 Observations With k-means clustering with $k = 4$, Clusters 1, 2, and 3 (shown in Tables 2, 3, and 4) had significantly lower max distances to center and sum squared error than Cluster 0 (Table 1). This set of clusters had the best overall fit to the data with Cluster 3 being less dense than the others. Agglomerative clustering with euclidean distance, centroid method, and a threshold of 12.00 produced identical clusters. In this case the best overall clustering was provided by agglomerative clustering since it did not require multiple runs to produce this ideal clustering where our implementation of k-means clustering required multiple runs due to the random cluster centroid selection. The results were identical, but it took less time to generate the agglomerative clustering.

3.2 mammal_milk

3.2.0.4 Best k-means clusters Initially $k = 3$ was used for k-means clustering of mammal_milk.csv since the file only contained 25 rows and there appeared to be a natural splits in the data of low, mid, and high ranges for attributes representing the percentages of water, protein, and fat. After several iterations with increasing k values and observing over-fitting at high k values, $k = 5$ was determined to give the best clustering with clear groupings based on the water percentage, as shown in Tables 9, 10, 11, 12, and 13.

Table 9: mammal_milk Cluster 0 data from k-means clustering, euclidean distance $k = 5$

Center	:				
	WATER	PROTEIN	FAT	LACTOSE	ASH
	65.16667	10.733334	20.400002	2.233333	1.2166666
Size	: 3				
Min Dist to Center	: 0.500278				
Max Dist to Center	: 1.201850				
Avg Dist to Center	: 0.948448				
Sum Squared Error	: 2.845345				
ANIMAL	WATER	PROTEIN	FAT	LACTOSE	ASH
Deer	65.9	10.4	19.7	2.6	1.4
Reindeer	64.8	10.7	20.3	2.5	1.4
Whale	64.8	11.1	21.2	1.6	0.85

Table 10: mammal_milk Cluster 1 data from k-means clustering, euclidean distance $k = 5$

Center	:				
	WATER	PROTEIN	FAT	LACTOSE	ASH
	82.0	7.116667	6.466668	4.183334	0.8866667
Size	: 6				
Min Dist to Center	: 1.124075				
Max Dist to Center	: 3.025505				
Avg Dist to Center	: 1.843383				
Sum Squared Error	: 11.060301				
ANIMAL	WATER	PROTEIN	FAT	LACTOSE	ASH
Buffalo	82.1	5.9	7.9	4.7	0.78
Guinea Pig	81.9	7.4	7.2	2.7	0.85
Cat	81.6	10.1	6.3	4.4	0.75
Fox	81.6	6.6	5.9	4.9	0.93
Pig	82.8	7.1	5.1	3.7	1.1
Sheep	82.0	5.6	6.4	4.7	0.91

Table 11: mammal_milk Cluster 2 data from k-means clustering, euclidean distance $k = 5$

Center	:				
	WATER	PROTEIN	FAT	LACTOSE	ASH
	45.65	10.14999999	38.45	0.45	0.69
Size	: 2				
Min Dist to Center	: 3.687221				
Max Dist to Center	: 3.687221				
Avg Dist to Center	: 3.687221				
Sum Squared Error	: 7.374442				
ANIMAL	WATER	PROTEIN	FAT	LACTOSE	ASH
Seal	46.4	9.7	42.0	0.0	0.85
Dolphin	44.9	10.6	34.9	0.9	0.53

Table 12: mammal_milk Cluster 3 data from k-means clustering, euclidean distance $k = 5$

Center	:				
	WATER	PROTEIN	FAT	LACTOSE	ASH
	88.5000001	2.57	2.8	5.68	0.485
Size	: 10				
Min Dist to Center	: 0.876541				
Max Dist to Center	: 3.488198				
Avg Dist to Center	: 2.307742				
Sum Squared Error	: 23.077424				
ANIMAL	WATER	PROTEIN	FAT	LACTOSE	ASH
Horse	90.1	2.6	1.0	6.9	0.35
Orangutan	88.5	1.4	3.5	6.0	0.24
Monkey	88.4	2.2	2.7	6.4	0.18
Donkey	90.3	1.7	1.4	6.2	0.4
Hippo	90.4	0.6	4.5	4.4	0.1
Camel	87.7	3.5	3.4	4.8	0.71
Bison	86.9	4.8	1.7	5.7	0.9
Llama	86.5	3.9	3.2	5.6	0.8
Mule	90.0	2.0	1.8	5.5	0.47
Zebra	86.2	3.0	4.8	5.3	0.7

Table 13: mammal_milk Cluster 4 data from k-means clustering, euclidean distance $k = 5$

Center	:				
	WATER	PROTEIN	FAT	LACTOSE	ASH
	72.7	8.6000001	13.2000001	3.45	1.38249998
Size	: 4				
Min Dist to Center	: 0.884763				
Max Dist to Center	: 7.317702				
Avg Dist to Center	: 4.445641				
Sum Squared Error	: 17.782563				
ANIMAL	WATER	PROTEIN	FAT	LACTOSE	ASH
Dog	76.3	9.3	9.5	3.0	1.2
Elephant	70.7	3.6	17.6	5.6	0.63
Rabbit	71.3	12.3	13.1	1.9	2.3
Rat	72.5	9.2	12.6	3.3	1.4

3.2.0.5 Best Agglomerative clusters Agglomerative clustering was initially run with euclidean distance, single link cluster distance, and no threshold to view the full tree structure and determine a good threshold for assigning clusters. It was found that using the centroid method with euclidean distance and a threshold of 10.00 produced the clusters with the best fit. Clusters are described in Tables 14, 15, 16, and 17.

Table 14: mammal_milk Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 1..00

Center	:				
	WATER	PROTEIN	FAT	LACTOSE	ASH
	45.65	10.1499999999	38.45	0.45	0.69
Size	: 2				
Min Dist to Center	: 3.687221				
Max Dist to Center	: 3.687221				
Avg Dist to Center	: 3.687221				
Sum Squared Error	: 7.374442				
ANIMAL	WATER	PROTEIN	FAT	LACTOSE	ASH
'Seal'	46.4	9.7	42.0	0.0	0.85
'Dolphin'	44.9	10.6	34.9	0.9	0.53

Table 15: mammal_milk Cluster 1 data from agglomerative clustering, euclidean distance centroid method threshold = 10.00

Center	:				
	WATER	PROTEIN	FAT	LACTOSE	ASH
	86.062499999	4.27499999995	4.175	5.11875	0.635625
Size	: 16				
Min Dist to Center	: 1.241482				
Max Dist to Center	: 7.673972				
Avg Dist to Center	: 4.510754				
Sum Squared Error	: 72.172070				
ANIMAL	WATER	PROTEIN	FAT	LACTOSE	ASH
'Bison'	86.9	4.8	1.7	5.7	0.9
'Zebra'	86.2	3.0	4.8	5.3	0.7
'Camel'	87.7	3.5	3.4	4.8	0.71
'Llama'	86.5	3.9	3.2	5.6	0.8
'Hippo'	90.4	0.6	4.5	4.4	0.1
'Orangutan'	88.5	1.4	3.5	6.0	0.24
'Monkey'	88.4	2.2	2.7	6.4	0.18
'Horse'	90.1	2.6	1.0	6.9	0.35
'Donkey'	90.3	1.7	1.4	6.2	0.4
'Mule'	90.0	2.0	1.8	5.5	0.47
'Cat'	81.6	10.1	6.3	4.4	0.75
'Guinea Pig'	81.9	7.4	7.2	2.7	0.85
'Pig'	82.8	7.1	5.1	3.7	1.1
'Buffalo'	82.1	5.9	7.9	4.7	0.78
'Fox'	81.6	6.6	5.9	4.9	0.93
'Sheep'	82.0	5.6	6.4	4.7	0.91

Table 16: mammal_milk Cluster 2 data from agglomerative clustering, euclidean distance centroid method threshold = 10.00

Center	:				
	WATER	PROTEIN	FAT	LACTOSE	ASH
	65.1666667	10.73333334	20.40000002	2.233333334	1.216666666
Size	: 3				
Min Dist to Center	: 0.500278				
Max Dist to Center	: 1.201850				
Avg Dist to Center	: 0.948448				
Sum Squared Error	: 2.845345				
ANIMAL	WATER	PROTEIN	FAT	LACTOSE	ASH
'Whale'	64.8	11.1	21.2	1.6	0.85
'Deer'	65.9	10.4	19.7	2.6	1.4
'Reindeer'	64.8	10.7	20.3	2.5	1.4

Table 17: mammal_milk Cluster 3 data from agglomerative clustering, euclidean distance centroid method threshold = 10.00

Center	:				
	WATER	PROTEIN	FAT	LACTOSE	ASH
	72.7	8.60000001	13.20000001	3.45	1.382499998
Size	: 4				
Min Dist to Center	: 0.884763				
Max Dist to Center	: 7.317702				
Avg Dist to Center	: 4.445641				
Sum Squared Error	: 17.782563				
ANIMAL	WATER	PROTEIN	FAT	LACTOSE	ASH
'Elephant'	70.7	3.6	17.6	5.6	0.63
'Dog'	76.3	9.3	9.5	3.0	1.2
'Rabbit'	71.3	12.3	13.1	1.9	2.3
'Rat'	72.5	9.2	12.6	3.3	1.4

3.2.0.6 Observations Using $k = 5$ for k-means clustering resulted in the best clustering results with clear groupings based on the water percentage, as shown in Tables 9, 10, 11, 12, and 13. By contrast, agglomerative clustering with the centroid method, euclidean distance, and a threshold of 10.00 only provided 4 clusters, but these clusters were nearly identical to those produced by the ideal k-means clustering. The "Deer, Reindeer, Whale", "Dog, Elephant, Rabbit, Rat", and "Seal, Dolphin" clusters occurred with both methods. The primary difference was that Clusters 1 and 3 for k-means clustering (Tables 10 and 12) were not differentiated based on their slightly different water percentages and very different protein percentages within agglomerative Cluster 1 (Table 15). The best clustering method in this case was again agglomerative since it did not require multiple runs to produce this nearly ideal clustering where our implementation of k-means clustering required multiple runs due to the random cluster centroid selection.

3.3 economy

3.3.0.7 Best k-means clusters Initially, $k = 2$ was used for economy.csv since it only contains 24 rows and there were no obvious splits in the data. After several iterations of k-means with random initial clusters and varying k values, $k = 6$ was determined to provide the best clustering. Those clusters are shown in Tables 18, 20, 22, 24, 26, 28.

Table 18: economy Cluster 0 data from k-means clustering, euclidean distance $k = 6$

Center	:									
	10.6	9.2	8.2	9.6	9.4	12.2	13.8	14.6	12.6	12.4
Size	: 5									
Min Dist to Center	: 1.385641									
Max Dist to Center	: 3.423449									
Avg Dist to Center	: 2.513106									
Sum Squared Error	: 12.565531									
'3'	10.0	9.0	8.0	10.0	10.0	12.0	14.0	14.0	12.0	12.0
'5'	13.0	10.0	9.0	10.0	10.0	11.0	14.0	15.0	13.0	12.0

Table 18: economy Cluster 0 data from k-means clustering, euclidean distance $k = 6$

Center :

	10.6	9.2	8.2	9.6	9.4	12.2	13.8	14.6	12.6	12.4

Size : 5
Min Dist to Center : 1.385641
Max Dist to Center : 3.423449
Avg Dist to Center : 2.513106
Sum Squared Error : 12.565531

'6'	10.0	8.0	8.0	9.0	10.0	13.0	14.0	15.0	13.0	12.0
'19'	9.0	8.0	7.0	9.0	8.0	12.0	13.0	13.0	12.0	13.0
'21'	11.0	11.0	9.0	10.0	9.0	13.0	14.0	16.0	13.0	13.0

Table 19: Cluster 0 data descriptions

3	:	Total curable
5	:	Electrical machinery equipment and supplie
6	:	Machinery except for electrica
19	:	Apparel and related product
21	:	Printing and publishing except newspapers

Table 20: economy Cluster 1 data from k-means clustering, euclidean distance $k = 6$

Center :

	9.75	8.0	7.0	7.0	7.5	9.25	10.0	10.25	8.25	9.0

Size : 4
Min Dist to Center : 2.549510
Max Dist to Center : 5.049752
Avg Dist to Center : 3.694159
Sum Squared Error : 14.776637

'8'	8.0	7.0	6.0	5.0	7.0	9.0	10.0	10.0	8.0	8.0
'10'	13.0	10.0	9.0	9.0	9.0	10.0	10.0	10.0	8.0	9.0
'18'	8.0	6.0	5.0	6.0	6.0	9.0	11.0	10.0	8.0	9.0
'20'	10.0	9.0	8.0	8.0	8.0	9.0	9.0	11.0	9.0	10.0

Table 21: Cluster 1 data descriptions

8	:	Primary iron and steel industry
10	:	Stone clay and glass products
18	:	Textile mill products
20	:	Paper and allied products

Table 22: economy Cluster 2 data from k-means clustering, euclidean distance $k = 6$

Center :

	13.5	13.0	11.0	14.0	14.5	15.5	19.0	18.5	15.0	16.0

Size : 2
Min Dist to Center : 5.567764
Max Dist to Center : 5.567764
Avg Dist to Center : 5.567764
Sum Squared Error : 11.135529

'4'	14.0	14.0	11.0	16.0	17.0	17.0	20.0	16.0	12.0	15.0
'13'	13.0	12.0	11.0	12.0	12.0	14.0	18.0	21.0	18.0	17.0

Table 23: Cluster 2 data descriptions

4	: Motor vehicles and equipment
13	: Instruments and related products

Table 24: economy Cluster 3 data from k-means clustering, euclidean distance $k = 6$

Center :

	9.834	9.334	9.5	9.666	9.666	11.0	11.834	12.666	11.834	11.834

Size : 6
Min Dist to Center : 1.462494
Max Dist to Center : 3.236081
Avg Dist to Center : 2.309425
Sum Squared Error : 13.856550

'2'	10.0	9.0	9.0	10.0	10.0	12.0	13.0	13.0	12.0	12.0
'14'	9.0	9.0	10.0	9.0	9.0	10.0	11.0	15.0	13.0	12.0
'15'	10.0	10.0	10.0	10.0	10.0	12.0	12.0	13.0	12.0	12.0
'16'	9.0	9.0	9.0	9.0	9.0	10.0	11.0	11.0	11.0	11.0
'23'	10.0	10.0	10.0	10.0	11.0	11.0	12.0	12.0	13.0	12.0
'24'	11.0	9.0	9.0	10.0	9.0	11.0	12.0	12.0	10.0	12.0

Table 25: Cluster 3 data descriptions

2	All manufacturing corporations except newspapers
14	Miscellaneous manufacture including ordnance
15	Total nondurable
16	Food and kindred products
23	Petroleum refining
24	Rubber and miscellaneous plastic products

Table 26: economy Cluster 4 data from k-means clustering, euclidean distance $k = 6$

Center	:									
	13.5	12.5	13.0	12.5	12.5	13.5	14.5	14.5	13.5	13.5
Size	: 2									
Min Dist to Center	: 1.802776									
Max Dist to Center	: 1.802776									
Avg Dist to Center	: 1.802776									
Sum Squared Error	: 3.605551									
'17'	13.0	13.0	14.0	13.0	13.0	13.0	14.0	14.0	14.0	14.0
'22'	14.0	12.0	12.0	12.0	12.0	14.0	15.0	15.0	13.0	13.0

Table 27: Cluster 4 data descriptions

17	: Tobacco manufacture
22	: Chemical and allied products

Table 28: economy Cluster 5 data from k-means clustering, euclidean distance $k = 6$

Center	:									
	8.6	6.0	5.2	7.4	7.8	10.2	12.0	13.4	11.4	12.6
Size	: 5									
Min Dist to Center	: 1.928730									
Max Dist to Center	: 5.892368									
Avg Dist to Center	: 3.155311									
Sum Squared Error	: 15.776553									
'7'	8.0	6.0	6.0	8.0	8.0	10.0	13.0	15.0	13.0	12.0
'9'	8.0	7.0	7.0	8.0	8.0	10.0	12.0	15.0	11.0	11.0
'11'	9.0	7.0	5.0	8.0	8.0	10.0	13.0	14.0	12.0	12.0
'12'	9.0	4.0	4.0	6.0	8.0	10.0	10.0	10.0	9.0	15.0
'25'	9.0	6.0	4.0	7.0	7.0	11.0	12.0	13.0	12.0	13.0

Table 29: Cluster 5 data descriptions

7	Fabricated metal products
9	Primary non-ferrous metal industry
11	Furniture and fixtures
12	Lumber and wood products except furniture
25	Leather and leather products

3.3.0.8 Best Agglomerative clusters Agglomerative clustering was initially run with euclidean distance, single link cluster distance, and no threshold to view the full tree structure and determine a good threshold for assigning clusters. It was found that using the centroid method with euclidean distance and a threshold of 9.00 produced the clusters with the best fit since it helped the "4" and "13" outlier rows move to their own clusters. The 4 resulting clusters are described in Tables 30, 32, 34, and 36.

Table 30: economy Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 9.00

Center :
9.7 8.2 7.6 8.55 8.7 10.75 12.0 12.85 11.2 11.6
Size : 20
Min Dist to Center : 2.601442
Max Dist to Center : 8.195578
Avg Dist to Center : 4.794304
Sum Squared Error : 95.886074

'12'	9.0	4.0	4.0	6.0	8.0	10.0	10.0	10.0	9.0	15.0
'25'	9.0	6.0	4.0	7.0	7.0	11.0	12.0	13.0	12.0	13.0
'9'	8.0	7.0	7.0	8.0	8.0	10.0	12.0	15.0	11.0	11.0
'7'	8.0	6.0	6.0	8.0	8.0	10.0	13.0	15.0	13.0	12.0
'11'	9.0	7.0	5.0	8.0	8.0	10.0	13.0	14.0	12.0	12.0
'5'	13.0	10.0	9.0	10.0	10.0	11.0	14.0	15.0	13.0	12.0
'21'	11.0	11.0	9.0	10.0	9.0	13.0	14.0	16.0	13.0	13.0
'14'	9.0	9.0	10.0	9.0	9.0	10.0	11.0	15.0	13.0	12.0
'19'	9.0	8.0	7.0	9.0	8.0	12.0	13.0	13.0	12.0	13.0
'6'	10.0	8.0	8.0	9.0	10.0	13.0	14.0	15.0	13.0	12.0
'2'	10.0	9.0	9.0	10.0	10.0	12.0	13.0	13.0	12.0	12.0
'3'	10.0	9.0	8.0	10.0	10.0	12.0	14.0	14.0	12.0	12.0
'15'	10.0	10.0	10.0	10.0	10.0	12.0	12.0	13.0	12.0	12.0
'23'	10.0	10.0	10.0	10.0	11.0	11.0	12.0	12.0	13.0	12.0
'16'	9.0	9.0	9.0	9.0	9.0	10.0	11.0	11.0	11.0	11.0
'24'	11.0	9.0	9.0	10.0	9.0	11.0	12.0	12.0	10.0	12.0
'8'	8.0	7.0	6.0	5.0	7.0	9.0	10.0	10.0	8.0	8.0
'18'	8.0	6.0	5.0	6.0	6.0	9.0	11.0	10.0	8.0	9.0
'10'	13.0	10.0	9.0	9.0	9.0	10.0	10.0	10.0	8.0	9.0
'20'	10.0	9.0	8.0	8.0	8.0	9.0	9.0	11.0	9.0	10.0

Table 31: Cluster 0 data description

2	All manufacturing corporations except newspapers
3	Total curable
5	Electrical machinery equipment and supplies
6	Machinery except for electrical
7	Fabricated metal products
8	Primary iron and steel industry
9	Primary non-ferrous metal industry
10	Stone clay and glass products
11	Furniture and fixtures
12	Lumber and wood products except furniture
14	Miscellaneous manufacture including ordnance
15	Total nondurable

Table 31: Cluster 0 data description

16	Food and kindred products
18	Textile mill products
19	Apparel and related products
20	Paper and allied products
21	Printing and publishing except newspapers
23	Petroleum refining
24	Rubber and miscellaneous plastic products
25	Leather and leather products

Table 32: economy Cluster 1 data from agglomerative clustering, euclidean distance centroid method threshold = 9.00

Center :										
	13.0	12.0	11.0	12.0	12.0	14.0	18.0	21.0	18.0	17.0
Size : 1										
Min Dist to Center : 0.000000										
Max Dist to Center : 0.000000										
Avg Dist to Center : 0.000000										
Sum Squared Error : 0.000000										
'13'	13.0	12.0	11.0	12.0	12.0	14.0	18.0	21.0	18.0	17.0

Table 33: Cluster 1 data description

13	Instruments and related products

Table 34: economy Cluster 2 data from agglomerative clustering, euclidean distance centroid method threshold = 9.00

Center :										
	14.0	14.0	11.0	16.0	17.0	17.0	20.0	16.0	12.0	15.0
Size : 1										
Min Dist to Center : 0.000000										
Max Dist to Center : 0.000000										
Avg Dist to Center : 0.000000										
Sum Squared Error : 0.000000										
'4'	14.0	14.0	11.0	16.0	17.0	17.0	20.0	16.0	12.0	15.0

Table 35: Cluster 2 data description

4	Motor vehicles and equipment

Table 36: economy Cluster 3 data from agglomerative clustering, euclidean distance centroid method threshold = 9.00

Center :
13.5 12.5 13.0 12.5 12.5 13.5 14.5 14.5 13.5 13.5
Size : 2
Min Dist to Center : 1.802776
Max Dist to Center : 1.802776
Avg Dist to Center : 1.802776
Sum Squared Error : 3.605551

'17'	13.0	13.0	14.0	13.0	13.0	13.0	14.0	14.0	14.0	14.0
'22'	14.0	12.0	12.0	12.0	12.0	14.0	15.0	15.0	13.0	13.0

Table 37: Cluster 3 data description

17	Tobacco manufacture
22	Chemical and allied products

3.3.0.9 Observations The economy data set contains the profit as a percentage of stockholder equity for various sectors of the economy during 1966. Both algorithms show certain clusters separating from the rest — 17 and 22, 4 and 13. These may be outliers, as the agglomerative clustering shows. They clearly stand out from the rest of the data in both algorithms, but 4 and 13 are singular in agglomerative. It is possible that motor "Motor vehicle equipment" and "Instruments and related products" would have distinct profit characteristics compared to the rest of the data.

On the other hand, 17 and 22, "Tobacco manufacture" and "Chemical and allied products" both cluster to each other in both clustering algorithms. It seems possible that stakeholders in one could hold stake in another, considering how many chemicals exist in tobacco products.

Investigating k-means, we see in Table 28 that two metal industries, a wood, a leather, and a furinture industry all get clustered together. It's possible that these industries go hand in hand to some degree, since all of these elements conceivably appeared in furniture in the 60's.

Looking at the clusters in Table 24, we find a lot of generic industries. These may be more run of the mill sectors of the economy that get similar levels of profit relative to their stakeholders — possibly stable investments. Interestingly, in Table 20, we see iron/steel, clay/glass, textile mill products, and paper products as a clustered sector. Iron/steel and clay/glass seem to go together — concrete famously uses rocks and steel rebar. Why textile mills and paper products made it into this cluster is interesting and somewhat strange. Textile mills take fiber and spin it into yarn to make clothes, while paper products are, well, unrelated, it seems. this seems like a strange clustering.

3.4 planets

3.4.0.10 Best k-means clusters Initially $k = 3$ was used for planets.csv since it only contains 19 rows and there appeared to be a natural split in the data under the Node attribute. After testing and evaluating several clustering attempts, $k = 4$ was found to fit the data best by clustering based on the Node attribute. These clusters are shown in Tables 38, 39, 40, and 41.

Table 38: planets Cluster 0 data from k-means clustering, euclidean distance $k = 4$

Center	:		
	Node	Inclination	Axis
	338.97900004	16.42	2.74
Size	: 2		
Min Dist to Center	: 0.736184		
Max Dist to Center	: 0.736184		
Avg Dist to Center	: 0.736184		
Sum Squared Error	: 1.472369		

Name	Node	Inclination	Axis
'1940YL'	338.333	16.773	2.7465
'1953NH'	339.625	16.067	2.7335

Table 39: planets Cluster 1 data from k-means clustering, euclidean distance $k = 4$

Center	:		
	Node	Inclination	Axis
	79.8416	4.528799995	2.68792
Size	: 5		
Min Dist to Center	: 1.087303		
Max Dist to Center	: 10.368646		
Avg Dist to Center	: 5.100430		
Sum Squared Error	: 25.502152		

Name	Node	Inclination	Axis
'1930SY'	80.804	4.622	2.189
'1949HM'	80.804	4.622	2.1906
'1948RO'	89.9	2.1	3.35
'1931DQ'	69.6	4.7	2.81
'1936AB'	78.1	6.6	2.9

Table 40: planets Cluster 2 data from k-means clustering, euclidean distance $k = 4$

Center	:		
	Node	Inclination	Axis
	141.192222	4.30900001	2.54563333
Size	: 9		
Min Dist to Center	: 6.498106		
Max Dist to Center	: 53.468784		
Avg Dist to Center	: 19.831158		
Sum Squared Error	: 178.480421		

Name	Node	Inclination	Axis
'1935RF'	130.916	4.659	2.2562
'1941FD'	132.2	4.7	2.13

Table 40: planets Cluster 2 data from k-means clustering, euclidean distance $k = 4$

Center	:		
	Node	Inclination	Axis
	141.192222	4.30900001	2.54563333
Size	: 9		
Min Dist to Center	: 6.498106		
Max Dist to Center	: 53.468784		
Avg Dist to Center	: 19.831158		
Sum Squared Error	: 178.480421		

Name	Node	Inclination	Axis
'1955QT'	130.07	4.79	2.1893
'1929EC'	115.072	2.666	3.1676
'1951AM'	115.072	2.666	3.1676
'1938DL'	135.6	1.0	2.6
'1951AX'	153.1	6.5	2.45
'1948RB'	194.6	1.8	3.02
'1948RH'	164.1	10.0	1.93

Table 41: planets Cluster 3 data from k-means clustering, euclidean distance $k = 4$

Center	:		
	Node	Inclination	Axis
	49.748	7.58066667	2.881433333
Size	: 3		
Min Dist to Center	: 6.194649		
Max Dist to Center	: 16.307787		
Avg Dist to Center	: 10.942523		
Sum Squared Error	: 32.827569		

Name	Node	Inclination	Axis
'1924TZ'	59.9	5.7	2.79
'1952DA'	55.144	4.542	3.0343
'1948TG'	34.2	12.5	2.82

3.4.0.11 Best Agglomerative clusters Agglomerative clustering was initially run with euclidean distance, single link cluster distance, and no threshold to view the full tree structure and determine a good threshold for assigning clusters. It was found that using the centroid method with euclidean distance and a threshold of 60.00 produced the 4 clusters with the best fit to the data. These clusters are shown in Tables 42, 43, 44, and 45.

Table 42: planets Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 60.00

Center	:		
	Node	Inclination	Axis
	338.97900000000004	16.42	2.74
Size	: 2		
Min Dist to Center	: 0.736184		
Max Dist to Center	: 0.736184		
Avg Dist to Center	: 0.736184		
Sum Squared Error	: 1.472369		
Name	Node	Inclination	Axis
'1940YL'	338.333	16.773	2.7465
'1953NH'	339.625	16.067	2.7335

Table 43: planets Cluster 1 data from agglomerative clustering, euclidean distance centroid method threshold = 60.00

Center	:		
	Node	Inclination	Axis
	68.5565	5.6732499999999995	2.7604875
Size	: 8		
Min Dist to Center	: 1.427781		
Max Dist to Center	: 35.028234		
Avg Dist to Center	: 14.303130		
Sum Squared Error	: 114.425039		
Name	Node	Inclination	Axis
'1948TG'	34.2	12.5	2.82
'1924TZ'	59.9	5.7	2.79
'1952DA'	55.144	4.542	3.0343
'1948RO'	89.9	2.1	3.35
'1931DQ'	69.6	4.7	2.81
'1936AB'	78.1	6.6	2.9
'1930SY'	80.804	4.622	2.189
'1949HM'	80.804	4.622	2.1906

Table 44: planets Cluster 2 data from agglomerative clustering, euclidean distance centroid method threshold = 60.00

Center	:		
	Node	Inclination	Axis
	194.6	1.8	3.02
Size	: 1		
Min Dist to Center	: 0.000000		
Max Dist to Center	: 0.000000		
Avg Dist to Center	: 0.000000		
Sum Squared Error	: 0.000000		
Name	Node	Inclination	Axis
'1948RB'	194.6	1.8	3.02

Table 45: planets Cluster 3 data from agglomerative clustering, euclidean distance centroid method threshold = 60.00

Center	:		
	Node	Inclination	Axis
	134.51624999999999	4.622625	2.4863375
Size	: 8		
Min Dist to Center	: 2.344777		
Max Dist to Center	: 30.073642		
Avg Dist to Center	: 12.756935		
Sum Squared Error	: 102.055480		
Name	Node	Inclination	Axis
'1951AX'	153.1	6.5	2.45
'1948RH'	164.1	10.0	1.93
'1929EC'	115.072	2.666	3.1676
'1951AM'	115.072	2.666	3.1676
'1938DL'	135.6	1.0	2.6
'1941FD'	132.2	4.7	2.13
'1935RF'	130.916	4.659	2.2562
'1955QT'	130.07	4.79	2.1893

3.4.0.12 Observations Running k-means clustering with $k = 4$ was found to fit the data best by clustering based on the Node attribute. These clusters are shown in Tables 38, 39, 40, and 41. With agglomerative clustering using euclidean distance, centroid method, and a threshold of 60.00, a few of the resulting clusters were similar to the k-means clusters. For example there was a "1940YL, 1953NH" cluster with both clustering methods. The k-means Clusters 1 and 3 (Tables 39 and 41) were combined in agglomerative clustering under Cluster 1 (Table 43). In contrast k-means Cluster 2 (Table 41) has it's Node value outlier "1948RB" split off into its own cluster under agglomerative clustering (Table 44). These differences are likely due to the random initial cluster centroid selection in k-means. Overall agglomerative gave better intuition for picking a threshold and produced reliable clusters faster than the random centroid k-means implementation.

3.5 iris

3.5.0.13 Best k-means clusters Based on the 3 classes of Iris in the data (Iris Setosa, Iris Versicolour, and Iris Virginica), $k = 3$ was selected initially for iris.csv despite it having 150 rows. Experimentation showed that the clustering results differed drastically based on the random initial cluster centroids, and that $k = 3$ was insufficient to differentiate all the classes of Iris. Using $k =$ 5 provided the most pure clusters of petal sizes and Iris classes, as shown in Tables 46, 47, 48, 49, and 50.

Table 46: Iris Cluster 0 data from k-means clustering, euclidean distance $k = 5$

Center	: 6.19655172413	2.882758620689	5.182758620689	1.9344827586206
Size	: 29			
Min Dist to Center	: 0.299544			
Max Dist to Center	: 0.852624			
Avg Dist to Center	: 0.528038			
Sum Squared Error	: 15.313097			
'Iris-versicolor'	5.9	3.2	4.8	1.8
'Iris-versicolor'	6.0	2.7	5.1	1.6
'Iris-virginica'	5.8	2.7	5.1	1.9
'Iris-virginica'	6.3	2.9	5.6	1.8
'Iris-virginica'	6.5	3.2	5.1	2.0
'Iris-virginica'	6.4	2.7	5.3	1.9
'Iris-virginica'	5.7	2.5	5.0	2.0
'Iris-virginica'	5.8	2.8	5.1	2.4
'Iris-virginica'	6.4	3.2	5.3	2.3
'Iris-virginica'	6.5	3.0	5.5	1.8
'Iris-virginica'	6.0	2.2	5.0	1.5
'Iris-virginica'	5.6	2.8	4.9	2.0
'Iris-virginica'	6.3	2.7	4.9	1.8
'Iris-virginica'	6.2	2.8	4.8	1.8
'Iris-virginica'	6.1	3.0	4.9	1.8
'Iris-virginica'	6.4	2.8	5.6	2.1
'Iris-virginica'	6.4	2.8	5.6	2.2
'Iris-virginica'	6.3	2.8	5.1	1.5
'Iris-virginica'	6.1	2.6	5.6	1.4
'Iris-virginica'	6.3	3.4	5.6	2.4
'Iris-virginica'	6.4	3.1	5.5	1.8
'Iris-virginica'	6.0	3.0	4.8	1.8
'Iris-virginica'	6.9	3.1	5.1	2.3
'Iris-virginica'	5.8	2.7	5.1	1.9
'Iris-virginica'	6.7	3.0	5.2	2.3
'Iris-virginica'	6.3	2.5	5.0	1.9
'Iris-virginica'	6.5	3.0	5.2	2.0
'Iris-virginica'	6.2	3.4	5.4	2.3
'Iris-virginica'	5.9	3.0	5.1	1.8

Table 47: Iris Cluster 1 data from k-means clustering, euclidean distance $k = 5$

Center	:			
	7.12272727274	3.113636363638	6.03181818181	2.13181818182
Size	: 22			
Min Dist to Center	: 0.178377			
Max Dist to Center	: 1.174347			
Avg Dist to Center	: 0.679665			
Sum Squared Error	: 14.952620			
'Iris-virginica'	6.3	3.3	6.0	2.5
'Iris-virginica'	7.1	3.0	5.9	2.1
'Iris-virginica'	6.5	3.0	5.8	2.2
'Iris-virginica'	7.6	3.0	6.6	2.1
'Iris-virginica'	7.3	2.9	6.3	1.8
'Iris-virginica'	6.7	2.5	5.8	1.8
'Iris-virginica'	7.2	3.6	6.1	2.5
'Iris-virginica'	6.8	3.0	5.5	2.1
'Iris-virginica'	7.7	3.8	6.7	2.2
'Iris-virginica'	7.7	2.6	6.9	2.3
'Iris-virginica'	6.9	3.2	5.7	2.3
'Iris-virginica'	7.7	2.8	6.7	2.0
'Iris-virginica'	6.7	3.3	5.7	2.1
'Iris-virginica'	7.2	3.2	6.0	1.8
'Iris-virginica'	7.2	3.0	5.8	1.6
'Iris-virginica'	7.4	2.8	6.1	1.9
'Iris-virginica'	7.9	3.8	6.4	2.0
'Iris-virginica'	7.7	3.0	6.1	2.3
'Iris-virginica'	6.9	3.1	5.4	2.1
'Iris-virginica'	6.7	3.1	5.6	2.4
'Iris-virginica'	6.8	3.2	5.9	2.3
'Iris-virginica'	6.7	3.3	5.7	2.5

Table 48: Iris Cluster 2 data from k-means clustering, euclidean distance $k = 5$

Center	:			
	6.399999999995	2.922727272724	4.58636363637	1.44090909091
Size	: 22			
Min Dist to Center	: 0.169528			
Max Dist to Center	: 0.757156			
Avg Dist to Center	: 0.453147			
Sum Squared Error	: 9.969227			
'Iris-versicolor'	7.0	3.2	4.7	1.4
'Iris-versicolor'	6.4	3.2	4.5	1.5
'Iris-versicolor'	6.9	3.1	4.9	1.5
'Iris-versicolor'	6.5	2.8	4.6	1.5
'Iris-versicolor'	6.3	3.3	4.7	1.6

Table 48: Iris Cluster 2 data from k-means clustering, euclidean distance $k = 5$

Center	:			
	6.399999999995	2.922727272724	4.58636363637	1.44090909091
Size	: 22			
Min Dist to Center	: 0.169528			
Max Dist to Center	: 0.757156			
Avg Dist to Center	: 0.453147			
Sum Squared Error	: 9.969227			
'Iris-versicolor'	6.6	2.9	4.6	1.3
'Iris-versicolor'	5.9	3.0	4.2	1.5
'Iris-versicolor'	6.1	2.9	4.7	1.4
'Iris-versicolor'	6.7	3.1	4.4	1.4
'Iris-versicolor'	6.2	2.2	4.5	1.5
'Iris-versicolor'	6.3	2.5	4.9	1.5
'Iris-versicolor'	6.1	2.8	4.7	1.2
'Iris-versicolor'	6.4	2.9	4.3	1.3
'Iris-versicolor'	6.6	3.0	4.4	1.4
'Iris-versicolor'	6.8	2.8	4.8	1.4
'Iris-versicolor'	6.7	3.0	5.0	1.7
'Iris-versicolor'	6.0	2.9	4.5	1.5
'Iris-versicolor'	6.0	3.4	4.5	1.6
'Iris-versicolor'	6.7	3.1	4.7	1.5
'Iris-versicolor'	6.3	2.3	4.4	1.3
'Iris-versicolor'	6.1	3.0	4.6	1.4
'Iris-versicolor'	6.2	2.9	4.3	1.3

Table 49: Iris Cluster 3 data from k-means clustering, euclidean distance $k = 5$

Center	:			
	5.0059999999	3.41800000006	1.464	0.24399999999
Size	: 50			
Min Dist to Center	: 0.059933			
Max Dist to Center	: 1.239351			
Avg Dist to Center	: 0.484132			
Sum Squared Error	: 24.206612			
'Iris-setosa'	5.1	3.5	1.4	0.2
'Iris-setosa'	4.9	3.0	1.4	0.2
'Iris-setosa'	4.7	3.2	1.3	0.2
'Iris-setosa'	4.6	3.1	1.5	0.2
'Iris-setosa'	5.0	3.6	1.4	0.2
'Iris-setosa'	5.4	3.9	1.7	0.4
'Iris-setosa'	4.6	3.4	1.4	0.3
'Iris-setosa'	5.0	3.4	1.5	0.2
'Iris-setosa'	4.4	2.9	1.4	0.2
'Iris-setosa'	4.9	3.1	1.5	0.1

Table 49: Iris Cluster 3 data from k-means clustering, euclidean distance $k = 5$

Center	:			
	5.0059999999	3.41800000006	1.464	0.24399999999
Size	: 50			
Min Dist to Center	: 0.059933			
Max Dist to Center	: 1.239351			
Avg Dist to Center	: 0.484132			
Sum Squared Error	: 24.206612			
'Iris-setosa'	5.4	3.7	1.5	0.2
'Iris-setosa'	4.8	3.4	1.6	0.2
'Iris-setosa'	4.8	3.0	1.4	0.1
'Iris-setosa'	4.3	3.0	1.1	0.1
'Iris-setosa'	5.8	4.0	1.2	0.2
'Iris-setosa'	5.7	4.4	1.5	0.4
'Iris-setosa'	5.4	3.9	1.3	0.4
'Iris-setosa'	5.1	3.5	1.4	0.3
'Iris-setosa'	5.7	3.8	1.7	0.3
'Iris-setosa'	5.1	3.8	1.5	0.3
'Iris-setosa'	5.4	3.4	1.7	0.2
'Iris-setosa'	5.1	3.7	1.5	0.4
'Iris-setosa'	4.6	3.6	1.0	0.2
'Iris-setosa'	5.1	3.3	1.7	0.5
'Iris-setosa'	4.8	3.4	1.9	0.2
'Iris-setosa'	5.0	3.0	1.6	0.2
'Iris-setosa'	5.0	3.4	1.6	0.4
'Iris-setosa'	5.2	3.5	1.5	0.2
'Iris-setosa'	5.2	3.4	1.4	0.2
'Iris-setosa'	4.7	3.2	1.6	0.2
'Iris-setosa'	4.8	3.1	1.6	0.2
'Iris-setosa'	5.4	3.4	1.5	0.4
'Iris-setosa'	5.2	4.1	1.5	0.1
'Iris-setosa'	5.5	4.2	1.4	0.2
'Iris-setosa'	4.9	3.1	1.5	0.1
'Iris-setosa'	5.0	3.2	1.2	0.2
'Iris-setosa'	5.5	3.5	1.3	0.2
'Iris-setosa'	4.9	3.1	1.5	0.1
'Iris-setosa'	4.4	3.0	1.3	0.2
'Iris-setosa'	5.1	3.4	1.5	0.2
'Iris-setosa'	5.0	3.5	1.3	0.3
'Iris-setosa'	4.5	2.3	1.3	0.3
'Iris-setosa'	4.4	3.2	1.3	0.2
'Iris-setosa'	5.0	3.5	1.6	0.6
'Iris-setosa'	5.1	3.8	1.9	0.4
'Iris-setosa'	4.8	3.0	1.4	0.3
'Iris-setosa'	5.1	3.8	1.6	0.2
'Iris-setosa'	4.6	3.2	1.4	0.2

Table 49: Iris Cluster 3 data from k-means clustering, euclidean distance $k = 5$

Center	:			
	5.0059999999	3.41800000006	1.464	0.24399999999
Size	: 50			
Min Dist to Center	: 0.059933			
Max Dist to Center	: 1.239351			
Avg Dist to Center	: 0.484132			
Sum Squared Error	: 24.206612			
'Iris-setosa'	5.3	3.7	1.5	0.2
'Iris-setosa'	5.0	3.3	1.4	0.2

Table 50: Iris Cluster 4 data from k-means clustering, euclidean distance $k = 5$

Center	:			
	5.518518517	2.6222222222	3.951851852	1.2185185188
Size	: 27			
Min Dist to Center	: 0.155688			
Max Dist to Center	: 1.053643			
Avg Dist to Center	: 0.530068			
Sum Squared Error	: 14.311849			
'Iris-versicolor'	5.5	2.3	4.0	1.3
'Iris-versicolor'	5.7	2.8	4.5	1.3
'Iris-versicolor'	4.9	2.4	3.3	1.0
'Iris-versicolor'	5.2	2.7	3.9	1.4
'Iris-versicolor'	5.0	2.0	3.5	1.0
'Iris-versicolor'	6.0	2.2	4.0	1.0
'Iris-versicolor'	5.6	2.9	3.6	1.3
'Iris-versicolor'	5.6	3.0	4.5	1.5
'Iris-versicolor'	5.8	2.7	4.1	1.0
'Iris-versicolor'	5.6	2.5	3.9	1.1
'Iris-versicolor'	6.1	2.8	4.0	1.3
'Iris-versicolor'	5.7	2.6	3.5	1.0
'Iris-versicolor'	5.5	2.4	3.8	1.1
'Iris-versicolor'	5.5	2.4	3.7	1.0
'Iris-versicolor'	5.8	2.7	3.9	1.2
'Iris-versicolor'	5.4	3.0	4.5	1.5
'Iris-versicolor'	5.6	3.0	4.1	1.3
'Iris-versicolor'	5.5	2.5	4.0	1.3
'Iris-versicolor'	5.5	2.6	4.4	1.2
'Iris-versicolor'	5.8	2.6	4.0	1.2
'Iris-versicolor'	5.0	2.3	3.3	1.0
'Iris-versicolor'	5.6	2.7	4.2	1.3
'Iris-versicolor'	5.7	3.0	4.2	1.2
'Iris-versicolor'	5.7	2.9	4.2	1.3
'Iris-versicolor'	5.1	2.5	3.0	1.1

Table 50: Iris Cluster 4 data from k-means clustering, euclidean distance $k = 5$

Center	:			
	5.518518517	2.6222222222	3.951851852	1.2185185188
Size	: 27			
Min Dist to Center	: 0.155688			
Max Dist to Center	: 1.053643			
Avg Dist to Center	: 0.530068			
Sum Squared Error	: 14.311849			
'Iris-versicolor'	5.7	2.8	4.1	1.3
'Iris-virginica'	4.9	2.5	4.5	1.7

3.5.0.14 Best Agglomerative clusters Agglomerative clustering was initially run with euclidean distance, single link cluster distance, and no threshold to view the full tree structure and determine a good threshold for assigning clusters. It was found that using the centroid method with euclidean distance and a threshold of 1.7 produced the 3 clusters with the best fit to the data. These clusters are shown in Tables 51, 52, and 53.

Table 51: Iris Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 1.7

Center	:			
	5.00599999	3.417999997	1.463999997	0.243999999
Size	: 50			
Min Dist to Center	: 0.059933			
Max Dist to Center	: 1.239351			
Avg Dist to Center	: 0.484132			
Sum Squared Error	: 24.206612			
'Iris-setosa'	4.5	2.3	1.3	0.3
'Iris-setosa'	5.7	4.4	1.5	0.4
'Iris-setosa'	5.8	4.0	1.2	0.2
'Iris-setosa'	5.4	3.9	1.7	0.4
'Iris-setosa'	5.7	3.8	1.7	0.3
'Iris-setosa'	5.4	3.9	1.3	0.4
'Iris-setosa'	5.2	4.1	1.5	0.1
'Iris-setosa'	5.5	4.2	1.4	0.2
'Iris-setosa'	4.6	3.6	1.0	0.2
'Iris-setosa'	4.3	3.0	1.1	0.1
'Iris-setosa'	4.4	3.2	1.3	0.2
'Iris-setosa'	4.4	2.9	1.4	0.2
'Iris-setosa'	4.4	3.0	1.3	0.2
'Iris-setosa'	4.8	3.4	1.6	0.2
'Iris-setosa'	4.8	3.4	1.9	0.2
'Iris-setosa'	5.0	3.2	1.2	0.2
'Iris-setosa'	4.7	3.2	1.6	0.2
'Iris-setosa'	4.8	3.1	1.6	0.2
'Iris-setosa'	5.0	3.0	1.6	0.2

Table 51: Iris Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 1.7

Center	:			
	5.00599999	3.417999997	1.463999997	0.243999999
Size	: 50			
Min Dist to Center	: 0.059933			
Max Dist to Center	: 1.239351			
Avg Dist to Center	: 0.484132			
Sum Squared Error	: 24.206612			
'Iris-setosa'	4.9	3.1	1.5	0.1
'Iris-setosa'	4.9	3.1	1.5	0.1
'Iris-setosa'	4.9	3.1	1.5	0.1
'Iris-setosa'	4.8	3.0	1.4	0.1
'Iris-setosa'	4.9	3.0	1.4	0.2
'Iris-setosa'	4.8	3.0	1.4	0.3
'Iris-setosa'	4.6	3.4	1.4	0.3
'Iris-setosa'	4.7	3.2	1.3	0.2
'Iris-setosa'	4.6	3.1	1.5	0.2
'Iris-setosa'	4.6	3.2	1.4	0.2
'Iris-setosa'	5.1	3.8	1.9	0.4
'Iris-setosa'	5.0	3.5	1.6	0.6
'Iris-setosa'	5.1	3.3	1.7	0.5
'Iris-setosa'	5.0	3.4	1.6	0.4
'Iris-setosa'	5.0	3.3	1.4	0.2
'Iris-setosa'	5.0	3.4	1.5	0.2
'Iris-setosa'	5.1	3.4	1.5	0.2
'Iris-setosa'	5.1	3.5	1.4	0.2
'Iris-setosa'	5.1	3.5	1.4	0.3
'Iris-setosa'	5.2	3.5	1.5	0.2
'Iris-setosa'	5.2	3.4	1.4	0.2
'Iris-setosa'	5.0	3.6	1.4	0.2
'Iris-setosa'	5.0	3.5	1.3	0.3
'Iris-setosa'	5.4	3.7	1.5	0.2
'Iris-setosa'	5.3	3.7	1.5	0.2
'Iris-setosa'	5.1	3.8	1.6	0.2
'Iris-setosa'	5.1	3.8	1.5	0.3
'Iris-setosa'	5.1	3.7	1.5	0.4
'Iris-setosa'	5.5	3.5	1.3	0.2
'Iris-setosa'	5.4	3.4	1.7	0.2
'Iris-setosa'	5.4	3.4	1.5	0.4

Table 52: Iris Cluster 1 data from agglomerative clustering, euclidean distance centroid method threshold = 1.7

Center	:			
	6.852777776	3.074999997	5.786111105	2.097222214
Size	: 36			
Min Dist to Center	: 0.257660			
Max Dist to Center	: 1.491736			
Avg Dist to Center	: 0.710720			
Sum Squared Error	: 25.585930			
'Iris-virginica'	6.3	3.3	6.0	2.5
'Iris-virginica'	6.4	3.2	5.3	2.3
'Iris-virginica'	6.3	3.4	5.6	2.4
'Iris-virginica'	6.2	3.4	5.4	2.3
'Iris-virginica'	6.7	3.3	5.7	2.1
'Iris-virginica'	6.9	3.2	5.7	2.3
'Iris-virginica'	6.8	3.2	5.9	2.3
'Iris-virginica'	6.7	3.1	5.6	2.4
'Iris-virginica'	6.7	3.3	5.7	2.5
'Iris-virginica'	6.5	3.0	5.8	2.2
'Iris-virginica'	6.4	2.8	5.6	2.1
'Iris-virginica'	6.4	2.8	5.6	2.2
'Iris-virginica'	6.4	2.7	5.3	1.9
'Iris-virginica'	6.3	2.9	5.6	1.8
'Iris-virginica'	6.5	3.0	5.5	1.8
'Iris-virginica'	6.4	3.1	5.5	1.8
'Iris-virginica'	6.5	3.2	5.1	2.0
'Iris-virginica'	6.5	3.0	5.2	2.0
'Iris-virginica'	6.8	3.0	5.5	2.1
'Iris-virginica'	6.9	3.1	5.4	2.1
'Iris-virginica'	6.9	3.1	5.1	2.3
'Iris-virginica'	6.7	3.0	5.2	2.3
'Iris-virginica'	6.7	2.5	5.8	1.8
'Iris-virginica'	6.1	2.6	5.6	1.4
'Iris-virginica'	7.7	2.6	6.9	2.3
'Iris-virginica'	7.6	3.0	6.6	2.1
'Iris-virginica'	7.7	2.8	6.7	2.0
'Iris-virginica'	7.7	3.0	6.1	2.3
'Iris-virginica'	7.3	2.9	6.3	1.8
'Iris-virginica'	7.4	2.8	6.1	1.9
'Iris-virginica'	7.1	3.0	5.9	2.1
'Iris-virginica'	7.2	3.2	6.0	1.8
'Iris-virginica'	7.2	3.0	5.8	1.6
'Iris-virginica'	7.2	3.6	6.1	2.5
'Iris-virginica'	7.7	3.8	6.7	2.2
'Iris-virginica'	7.9	3.8	6.4	2.0

Table 53: Iris Cluster 2 data from agglomerative clustering, euclidean distance centroid method threshold = 1.7

Center	:			
	5.9296875001	2.7578125001	4.4109375	1.4390625
Size	: 64			
Min Dist to Center	: 0.191851			
Max Dist to Center	: 1.691318			
Avg Dist to Center	: 0.747891			
Sum Squared Error	: 47.865013			
'Iris-versicolor'	5.0	2.0	3.5	1.0
'Iris-versicolor'	5.1	2.5	3.0	1.1
'Iris-versicolor'	4.9	2.4	3.3	1.0
'Iris-versicolor'	5.0	2.3	3.3	1.0
'Iris-virginica'	4.9	2.5	4.5	1.7
'Iris-versicolor'	6.0	2.2	4.0	1.0
'Iris-versicolor'	5.6	3.0	4.5	1.5
'Iris-versicolor'	5.4	3.0	4.5	1.5
'Iris-versicolor'	5.9	3.0	4.2	1.5
'Iris-versicolor'	5.7	2.8	4.5	1.3
'Iris-versicolor'	5.5	2.6	4.4	1.2
'Iris-versicolor'	5.6	3.0	4.1	1.3
'Iris-versicolor'	5.7	3.0	4.2	1.2
'Iris-versicolor'	5.7	2.9	4.2	1.3
'Iris-versicolor'	5.6	2.7	4.2	1.3
'Iris-versicolor'	5.7	2.8	4.1	1.3
'Iris-versicolor'	5.8	2.7	4.1	1.0
'Iris-versicolor'	5.8	2.7	3.9	1.2
'Iris-versicolor'	5.8	2.6	4.0	1.2
'Iris-versicolor'	5.2	2.7	3.9	1.4
'Iris-versicolor'	5.5	2.3	4.0	1.3
'Iris-versicolor'	5.5	2.5	4.0	1.3
'Iris-versicolor'	5.6	2.5	3.9	1.1
'Iris-versicolor'	5.5	2.4	3.8	1.1
'Iris-versicolor'	5.5	2.4	3.7	1.0
'Iris-versicolor'	5.6	2.9	3.6	1.3
'Iris-versicolor'	5.7	2.6	3.5	1.0
'Iris-virginica'	6.0	2.2	5.0	1.5
'Iris-versicolor'	6.2	2.2	4.5	1.5
'Iris-versicolor'	6.3	2.3	4.4	1.3
'Iris-virginica'	5.8	2.8	5.1	2.4
'Iris-virginica'	5.6	2.8	4.9	2.0
'Iris-virginica'	5.7	2.5	5.0	2.0
'Iris-virginica'	5.8	2.7	5.1	1.9
'Iris-virginica'	5.8	2.7	5.1	1.9
'Iris-virginica'	5.9	3.0	5.1	1.8
'Iris-versicolor'	5.9	3.2	4.8	1.8
'Iris-virginica'	6.1	3.0	4.9	1.8

Table 53: Iris Cluster 2 data from agglomerative clustering, euclidean distance centroid method threshold = 1.7

Center	:			
	5.9296875001	2.7578125001	4.4109375	1.4390625
Size	: 64			
Min Dist to Center	: 0.191851			
Max Dist to Center	: 1.691318			
Avg Dist to Center	: 0.747891			
Sum Squared Error	: 47.865013			
'Iris-virginica'	6.0	3.0	4.8	1.8
'Iris-virginica'	6.3	2.5	5.0	1.9
'Iris-virginica'	6.3	2.7	4.9	1.8
'Iris-virginica'	6.2	2.8	4.8	1.8
'Iris-versicolor'	6.3	2.5	4.9	1.5
'Iris-versicolor'	6.0	2.7	5.1	1.6
'Iris-virginica'	6.3	2.8	5.1	1.5
'Iris-versicolor'	6.7	3.1	4.4	1.4
'Iris-versicolor'	6.6	3.0	4.4	1.4
'Iris-versicolor'	6.5	2.8	4.6	1.5
'Iris-versicolor'	6.6	2.9	4.6	1.3
'Iris-versicolor'	6.7	3.0	5.0	1.7
'Iris-versicolor'	6.8	2.8	4.8	1.4
'Iris-versicolor'	6.7	3.1	4.7	1.5
'Iris-versicolor'	7.0	3.2	4.7	1.4
'Iris-versicolor'	6.9	3.1	4.9	1.5
'Iris-versicolor'	6.0	3.4	4.5	1.6
'Iris-versicolor'	6.4	3.2	4.5	1.5
'Iris-versicolor'	6.3	3.3	4.7	1.6
'Iris-versicolor'	6.1	2.8	4.7	1.2
'Iris-versicolor'	6.0	2.9	4.5	1.5
'Iris-versicolor'	6.1	2.9	4.7	1.4
'Iris-versicolor'	6.1	3.0	4.6	1.4
'Iris-versicolor'	6.1	2.8	4.0	1.3
'Iris-versicolor'	6.4	2.9	4.3	1.3
'Iris-versicolor'	6.2	2.9	4.3	1.3

3.5.0.15 Observations Running k-means with $k = 5$ on the iris data set resulted in clusters that were consistent in size and nearly pure by iris class. The exceptions to this were Cluster 0 (Table 46), which had 2 Iris-versicolor in a predominantly Iris-virginica cluster, and Cluster 4 (Table 50), which had 1 Iris-virginica in a predominantly Iris-versicolor cluster. Achieving this close to ideal clustering took many attempts, however, due to the random selection of initial cluster centroids. In contrast agglomerative clustering with euclidean distance, centroid method, and a threshold of 1.7 had only 3 clusters, and of those only Cluster 2 (Table 53) had other classes of Iris mixed in (14 Iris-virginica in a predominantly Iris-versicolor). Despite the utility of the separation within Iris classes found with the k-means clustering at $k = 5$, the number of iterations it took to generate those clusters with random initial centroid selection overall made it perform less well than agglomerative clustering.

3.6 many_clusters

3.6.0.16 Best k-means clusters Initially $k = 6$ was selected for the many_clusters.csv since it contained 73 data points and plotting the points on a 2D plane showed that it contained at least 6 loose clusters. Since initial cluster centroids are randomly selected, it took a few tries to get the $k = 6$ clusters shown in Tables 54, 55, 56, 57, 58, and 59.

Table 54: many_clusters Cluster 0 data from k-means clustering, euclidean distance $k = 6$

Center	:
8.785714285714286	33.642857142857146
Size	: 14
Min Dist to Center	: 1.265718
Max Dist to Center	: 11.971480
Avg Dist to Center	: 5.028803
Sum Squared Error	: 70.403241
10.0	41.0
3.0	38.0
6.0	37.0
8.0	37.0
7.0	36.0
13.0	36.0
6.0	35.0
8.0	35.0
10.0	34.0
5.0	33.0
9.0	32.0
11.0	28.0
18.0	26.0
9.0	23.0

Table 55: many_clusters Cluster 1 data from k-means clustering, euclidean distance $k = 6$

Center	:
41.27272727272727	38.90909090909091
Size	: 11
Min Dist to Center	: 1.676281
Max Dist to Center	: 10.085584
Avg Dist to Center	: 4.396795
Sum Squared Error	: 48.364744
38.0	45.0
42.0	43.0
40.0	42.0
41.0	41.0
44.0	41.0
40.0	40.0

Table 55: many_clusters Cluster 1 data from k-means clustering, euclidean distance $k = 6$

Center	:
41.27272727272727	38.90909090909091
Size	: 11
Min Dist to Center	: 1.676281
Max Dist to Center	: 10.085584
Avg Dist to Center	: 4.396795
Sum Squared Error	: 48.364744
44.0	40.0
43.0	37.0
38.0	36.0
38.0	33.0
46.0	30.0

Table 56: many_clusters Cluster 2 data from k-means clustering, euclidean distance $k = 6$

Center	:
23.3	41.7
Size	: 10
Min Dist to Center	: 0.424264
Max Dist to Center	: 9.109336
Avg Dist to Center	: 4.005880
Sum Squared Error	: 40.058800
24.0	49.0
19.0	44.0
23.0	44.0
22.0	43.0
24.0	43.0
26.0	43.0
23.0	42.0
20.0	39.0
26.0	37.0
26.0	33.0

Table 57: many_clusters Cluster 3 data from k-means clustering, euclidean distance $k = 6$

Center	:
11.357142857142858	8.0
Size	: 14
Min Dist to Center	: 1.061862
Max Dist to Center	: 10.862076
Avg Dist to Center	: 4.972720
Sum Squared Error	: 69.618085
11.0	18.0
7.0	14.0
21.0	13.0
12.0	10.0
16.0	9.0
5.0	8.0
9.0	7.0
11.0	7.0
10.0	6.0
13.0	6.0
9.0	5.0
11.0	5.0
10.0	3.0
14.0	1.0

Table 58: many_clusters Cluster 4 data from k-means clustering, euclidean distance $k = 6$

Center	:
41.2	9.6
Size	: 10
Min Dist to Center	: 2.607681
Max Dist to Center	: 15.658863
Avg Dist to Center	: 6.250852
Sum Squared Error	: 62.508518
50.0	19.0
39.0	16.0
44.0	15.0
39.0	11.0
44.0	8.0
42.0	7.0
41.0	6.0
43.0	6.0
43.0	5.0
27.0	3.0

Table 59: many_clusters Cluster 5 data from k-means clustering, euclidean distance $k = 6$

Center	:
34.0	24.5
Size	: 14
Min Dist to Center	: 1.118034
Max Dist to Center	: 12.298374
Avg Dist to Center	: 5.321519
Sum Squared Error	: 74.501268
31.0	30.0
35.0	30.0
28.0	28.0
34.0	27.0
36.0	27.0
37.0	26.0
35.0	25.0
39.0	25.0
37.0	24.0
41.0	23.0
31.0	21.0
38.0	21.0
23.0	19.0
31.0	17.0

3.6.0.17 Best Agglomerative clusters

Agglomerative clustering was initially run with euclidean distance, single link cluster distance, and no threshold to view the full tree structure and determine a good threshold for assigning clusters. It was found that using the centroid method with euclidean distance and a threshold of 14.00 produced the 6 clusters with the best fit to the data. These clusters are shown in Tables 60, 61, 62, 63, 64, and 65.

Table 60: many_clusters Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 14.00

Center	:
13.0625	8.375
Size	: 16
Min Dist to Center	: 1.941528
Max Dist to Center	: 14.938023
Avg Dist to Center	: 6.604921
Sum Squared Error	: 105.678736
11.0	18.0
7.0	14.0
5.0	8.0
14.0	1.0
10.0	3.0
13.0	6.0

Table 60: many_clusters Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 14.00

Center	:
13.0625	8.375
Size	: 16
Min Dist to Center	: 1.941528
Max Dist to Center	: 14.938023
Avg Dist to Center	: 6.604921
Sum Squared Error	: 105.678736
11.0	5.0
9.0	5.0
11.0	7.0
9.0	7.0
10.0	6.0
12.0	10.0
16.0	9.0
27.0	3.0
23.0	19.0
21.0	13.0

Table 61: many_clusters Cluster 1 data from agglomerative clustering, euclidean distance centroid method threshold = 14.00

Center	:
22.625	43.375
Size	: 8
Min Dist to Center	: 0.728869
Max Dist to Center	: 5.790617
Avg Dist to Center	: 2.784391
Sum Squared Error	: 22.275131
24.0	49.0
20.0	39.0
19.0	44.0
26.0	43.0
23.0	44.0
22.0	43.0
24.0	43.0
23.0	42.0

Table 62: many_clusters Cluster 2 data from agglomerative clustering, euclidean distance centroid method threshold = 14.00

Center	:
8.785714285714286	33.642857142857146
Size	: 14
Min Dist to Center	: 1.265718
Max Dist to Center	: 11.971480
Avg Dist to Center	: 5.028803
Sum Squared Error	: 70.403241
3.0	38.0
10.0	34.0
9.0	32.0
5.0	33.0
8.0	35.0
6.0	35.0
8.0	37.0
6.0	37.0
7.0	36.0
10.0	41.0
13.0	36.0
18.0	26.0
11.0	28.0
9.0	23.0

Table 63: many_clusters Cluster 3 data from agglomerative clustering, euclidean distance centroid method threshold = 14.00

Center	:
42.77777777777778	10.333333333333334
Size	: 9
Min Dist to Center	: 2.634060
Max Dist to Center	: 11.281472
Avg Dist to Center	: 5.241103
Sum Squared Error	: 47.169924
50.0	19.0
44.0	8.0
43.0	6.0
43.0	5.0
42.0	7.0
41.0	6.0
44.0	15.0
39.0	16.0
39.0	11.0

Table 64: many_clusters Cluster 4 data from agglomerative clustering, euclidean distance centroid method threshold = 14.00

Center	:
41.27272727272727	38.90909090909091
Size	: 11
Min Dist to Center	: 1.676281
Max Dist to Center	: 10.085584
Avg Dist to Center	: 4.396795
Sum Squared Error	: 48.364744
46.0	30.0
38.0	36.0
38.0	33.0
38.0	45.0
43.0	37.0
44.0	41.0
44.0	40.0
42.0	43.0
40.0	40.0
40.0	42.0
41.0	41.0

Table 65: many_clusters Cluster 5 data from agglomerative clustering, euclidean distance centroid method threshold = 14.00

Center	:
33.666666666666664	26.266666666666666
Size	: 15
Min Dist to Center	: 0.805536
Max Dist to Center	: 13.190232
Avg Dist to Center	: 5.747558
Sum Squared Error	: 86.213364
31.0	30.0
28.0	28.0
26.0	37.0
26.0	33.0
31.0	21.0
31.0	17.0
41.0	23.0
38.0	21.0
35.0	30.0
34.0	27.0
35.0	25.0
36.0	27.0
37.0	26.0
39.0	25.0

Table 65: many_clusters Cluster 5 data from agglomerative clustering, euclidean distance centroid method threshold = 14.00

Center	:
33.666666666666664	26.266666666666666
Size	: 15
Min Dist to Center	: 0.805536
Max Dist to Center	: 13.190232
Avg Dist to Center	: 5.747558
Sum Squared Error	: 86.213364
37.0	24.0

3.6.0.18 Observations Running k-means with $k = 6$ on the many_clusters data set took a few attempts before generating 6 clusters that fit the fairly low average distance to the center, as shown in Tables 54, 55, 56, 57, 58, and 59. Agglomerative clustering using the centroid method with euclidean distance and a threshold of 14.00 also produced the 6 clusters that fit the data well, and 2 of the agglomerative clusters matched the k-means clusters exactly (k-means Cluster 0 to agglomerative Cluster 2 in Tables 54 and 62, and k-means Cluster 1 to agglomerative Cluster 4 in Tables 55 and 64). The other 4 clusters had fairly similar cluster centers despite containing slightly different combinations of tuples, as shown in Table 66.

Table 66: many_clusters comparison of cluster centers in k-means and agglomerative clusterings

k-means			agglomerative		
Cluster 2	23.3	41.7	Cluster 1	22.625	43.375
Cluster 3	11.3571428571	8.0	Cluster 0	13.0625	8.375
Cluster 4	41.2	9.6	Cluster 3	42.777777778	10.3333333334
Cluster 5	34.0	24.5	Cluster 5	33.6666666664	26.2666666666

Overall the agglomerative clustering method was better for than k-means with random intial clusters because it did not need many iterations to produce clusters that fit the data well.

3.7 AccidentsSet01

3.7.0.19 Best k-means clusters Initially $k = 2$ was used for AccidentsSet01.csv since it only contains 19 rows and there appeared to be a natural split in the data on VE_TOTAL, the number of vehicles in the accident attribute. This held true through several runs of k-means clustering despite the random selection of cluster centroids. However, the large range in the PERSONS attribute within those clusters led us to trying $k = 3$, which reduced the sum squared error and overall fit the data much better since it resulted in splitting the cluster with a large number of vehicles and a large range of PERSONS values. The best clusters from using k =3 are shown in Tables 67, 68, and 69 which represent many-vehicle accidents with $>$ 11 persons, 2 vehicle accidents with $<$ 6 persons, and many-vehicle accidents with $<=$ 11 persons respectively. All data points were successfully clustered and the number of data points in each cluster was fairly balanced.

Table 67: AccidentsSet01 Cluster 0 data from k-means clustering, euclidean distance $k = 3$

Center	:	
VE_TOTAL	PERSONS	FATALS
4.25	15.0	1.25
Size	: 4	
Min Dist to Center	: 0.790569	
Max Dist to Center	: 4.6502694	
Avg Dist to Center	: 2.454503	
Sum Squared Error	: 9.818012	
VE_TOTAL	PERSONS	FATALS
5.0	15.0	1.0
5.0	14.0	1.0
5.0	12.0	1.0
2.0	19.0	2.0

Table 68: AccidentsSet01 Cluster 1 data from k-means clustering, euclidean distance $k = 3$

Center	:	
VE_TOTAL	PERSONS	FATALS
2.0	3.125	2.0
Size	: 8	
Min Dist to Center	: 0.875000	
Max Dist to Center	: 2.741464	
Avg Dist to Center	: 1.646746	
Sum Squared Error	: 13.173968	
VE_TOTAL	PERSONS	FATALS
2.0	1.0	1.0
2.0	2.0	2.0
2.0	2.0	2.0
2.0	2.0	1.0
2.0	4.0	2.0
2.0	4.0	1.0
2.0	5.0	3.0
2.0	5.0	4.0

Table 69: AccidentsSet01 Cluster 2 data from k-means clustering, euclidean distance $k = 3$

Center	:	
VE_TOTAL	PERSONS	FATALS
5.0	8.857142857142858	1.0
Size	: 7	
Min Dist to Center	: 0.142857	
Max Dist to Center	: 2.142857	
Avg Dist to Center	: 1.020408	
Sum Squared Error	: 7.142857	
VE_TOTAL	PERSONS	FATALS
5.0	11.0	1.0
5.0	10.0	1.0
5.0	9.0	1.0
5.0	9.0	1.0
5.0	8.0	1.0
5.0	8.0	1.0
5.0	7.0	1.0

3.7.0.20 Best Agglomerative clusters Agglomerative clustering was initially run with euclidean distance, single link cluster distance, and no threshold to view the full tree structure and determine a good threshold for assigning clusters. It was found that using the centroid method with euclidean distance and a threshold of 5.00 produced the 3 clusters with the best fit to the data. These clusters are shown in Tables 70, 71, and 72.

Table 70: AccidentsSet01 Cluster 0 data from euclidean distance centroid method threshold = 5.00)

Center	:	
VE_TOTAL	PERSONS	FATALS
2.0	19.0	2.0
Size	: 1	
Min Dist to Center	: 0.000000	
Max Dist to Center	: 0.000000	
Avg Dist to Center	: 0.000000	
Sum Squared Error	: 0.000000	
VE_TOTAL	PERSONS	FATALS
2.0	19.0	2.0

Table 71: AccidentsSet01 Cluster 1 data from euclidean distance centroid method threshold = 5.00)

Center	:	
VE_TOTAL	PERSONS	FATALS
2.0	3.125	2.0
Size	: 8	
Min Dist to Center	: 0.875000	
Max Dist to Center	: 2.741464	
Avg Dist to Center	: 1.646746	
Sum Squared Error	: 13.173968	
VE_TOTAL	PERSONS	FATALS
2.0	2.0	2.0
2.0	2.0	2.0
2.0	1.0	1.0
2.0	2.0	1.0
2.0	4.0	2.0
2.0	4.0	1.0
2.0	5.0	3.0
2.0	5.0	4.0

Table 72: AccidentsSet01 Cluster 2 data from euclidean distance centroid method threshold = 5.00)

Center	:	
VE_TOTAL	PERSONS	FATALS
5.0	10.3	1.0
Size	: 10	
Min Dist to Center	: 0.300000	
Max Dist to Center	: 4.700000	
Avg Dist to Center	: 2.160000	
Sum Squared Error	: 21.600000	
VE_TOTAL	PERSONS	FATALS
5.0	10.0	1.0
5.0	9.0	1.0
5.0	9.0	1.0
5.0	7.0	1.0
5.0	8.0	1.0
5.0	8.0	1.0
5.0	15.0	1.0
5.0	14.0	1.0
5.0	12.0	1.0
5.0	11.0	1.0

3.7.0.21 Observations Running k-means clustering with $k = 2$ resulted in 2 clusters split on the number of vehicles attribute. Increasing the number of clusters to $k = 3$ resulted in 3 clusters split on the number of persons and the number of vehicles, and was determined to be the best

k-means clustering for the data. In contrast the agglomerative clustering split almost entirely based on VE_TOTAL, and the outlier 2.0 car accident with 19 PERSONS involved was placed in it's own cluster (Cluster 0 in Table 70) where in the k-means clustering it was incorporated in a cluster with 5 VE_TOTAL accidents that also had a high number of PERSONS involved. Again, overall agglomerative clustering performed better overall by sectioning the outlier data point into its own cluster and not having the variability from random initial cluster centroids that existed in our k-means implementation.

3.8 AccidentsSet02

3.8.0.22 Best k-means clusters Initially $k = 3$ was used for AccidentsSet02.csv since it only contains 49 rows and there appeared to be a natural split in the data on SP_LIMIT attribute with values of 35.00, 45.00, and 70.00. This clustering did result after a few attempts due to the random selection of cluster centroids. Looking at the $k = 3$ clusters, there also seemed to be possible splitting based on NO_LANES in 2 of the clusters. After several attempts to get better initial cluster centroids, $k = 5$ was found to produce very balanced clusters of the data, as shown in Tables 73, 74, 75, 76, and 77.

Table 73: AccidentsSet02 Cluster 0 data from k-means clustering, euclidean distance $k = 5$

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.2	1.6	0.6	9.0	35.0	1.0	0.0
Size	: 5					
Min Dist to Center	: 0.600000					
Max Dist to Center	: 1.166190					
Avg Dist to Center	: 0.767594					
Sum Squared Error	: 3.837970					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.0	2.0	1.0	9.0	35.0	1.0	0.0
1.0	2.0	1.0	9.0	35.0	1.0	0.0
2.0	1.0	0.0	9.0	35.0	1.0	0.0
1.0	2.0	1.0	9.0	35.0	1.0	0.0
1.0	1.0	0.0	9.0	35.0	1.0	0.0

Table 74: AccidentsSet02 Cluster 1 data from k-means clustering, euclidean distance $k = 5$

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.666667	2.333335	0.3333	6.16667	35.0	1.0	0.0
Size	: 6					
Min Dist to Center	: 0.600925					
Max Dist to Center	: 1.536591					
Avg Dist to Center	: 1.162636					
Sum Squared Error	: 6.975815					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.0	2.0	1.0	7.0	35.0	1.0	0.0
1.0	1.0	0.0	6.0	35.0	1.0	0.0
2.0	2.0	0.0	6.0	35.0	1.0	0.0
1.0	3.0	1.0	6.0	35.0	1.0	0.0
3.0	3.0	0.0	6.0	35.0	1.0	0.0
2.0	3.0	0.0	6.0	35.0	1.0	0.0

Table 75: AccidentsSet02Cluster 2 data from k-means clustering, euclidean distance $k = 5$

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.73333334	4.5333333	0.0	2.0	70.0	1.26666666	0.33333333
Size	: 15					
Min Dist to Center	: 1.415784					
Max Dist to Center	: 20.607550					
Avg Dist to Center	: 4.447912					
Sum Squared Error	: 66.718682					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.0	1.0	0.0	2.0	70.0	1.0	0.0
2.0	2.0	0.0	2.0	70.0	1.0	0.0
6.0	9.0	0.0	2.0	70.0	2.0	0.0
1.0	1.0	0.0	2.0	70.0	1.0	0.0
4.0	25.0	0.0	2.0	70.0	2.0	0.0
1.0	1.0	0.0	2.0	70.0	1.0	0.0
1.0	1.0	0.0	2.0	70.0	1.0	0.0
1.0	3.0	0.0	2.0	70.0	1.0	0.0
1.0	6.0	0.0	2.0	70.0	3.0	2.0
1.0	1.0	0.0	2.0	70.0	1.0	0.0
1.0	8.0	0.0	2.0	70.0	1.0	0.0
1.0	2.0	0.0	2.0	70.0	1.0	2.0
3.0	5.0	0.0	2.0	70.0	1.0	0.0
1.0	2.0	0.0	2.0	70.0	1.0	0.0
1.0	1.0	0.0	2.0	70.0	1.0	1.0

Table 76: AccidentsSet02Cluster 3 data from k-means clustering, euclidean distance $k = 5$

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.72727273	5.0909091	0.5454	3.36363638	35.0	1.363635	0.454545453
Size	: 11					
Min Dist to Center	: 1.102214					
Max Dist to Center	: 4.684245					
Avg Dist to Center	: 2.213141					
Sum Squared Error	: 24.344546					

VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
3.0	4.0	0.0	4.0	35.0	1.0	1.0
2.0	6.0	0.0	4.0	35.0	1.0	0.0
3.0	4.0	0.0	4.0	35.0	2.0	0.0
1.0	4.0	0.0	4.0	35.0	3.0	1.0
2.0	4.0	0.0	4.0	35.0	1.0	1.0
1.0	4.0	0.0	4.0	35.0	1.0	0.0
1.0	5.0	1.0	3.0	35.0	1.0	0.0
1.0	6.0	5.0	3.0	35.0	2.0	0.0
2.0	6.0	0.0	3.0	35.0	1.0	1.0
1.0	4.0	0.0	2.0	35.0	1.0	0.0
2.0	9.0	0.0	2.0	35.0	1.0	1.0

Table 77: AccidentsSet02Cluster 4 data from k-means clustering, euclidean distance $k = 5$

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.41666667	3.08333335	0.91666	4.0	45.0	1.16667	0.41666667
Size	: 12					
Min Dist to Center	: 0.623610					
Max Dist to Center	: 4.801620					
Avg Dist to Center	: 1.754894					
Sum Squared Error	: 21.058727					

VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.0	3.0	1.0	4.0	45.0	1.0	1.0
1.0	7.0	0.0	4.0	45.0	1.0	0.0
1.0	2.0	1.0	4.0	45.0	1.0	1.0
1.0	2.0	1.0	4.0	45.0	1.0	0.0
2.0	2.0	0.0	4.0	45.0	1.0	1.0
1.0	2.0	1.0	4.0	45.0	1.0	1.0
1.0	2.0	1.0	4.0	45.0	1.0	1.0
1.0	3.0	2.0	4.0	45.0	2.0	0.0
1.0	2.0	1.0	4.0	45.0	1.0	0.0
1.0	3.0	1.0	4.0	45.0	1.0	0.0
5.0	6.0	0.0	4.0	45.0	2.0	0.0
1.0	3.0	2.0	4.0	45.0	1.0	0.0

3.8.0.23 Best Agglomerative clusters Agglomerative clustering was initially run with euclidean distance, single link cluster distance, and no threshold to view the full tree structure and determine a good threshold for assigning clusters. It was found that using the centroid method with euclidean distance and a threshold of 10.00 produced the 4 clusters with the best fit to the data by separating on SP_LIMIT and separating the outlier with a much higher PERSONS value into its own cluster. These clusters are shown in Tables 78 .

Table 78: AccidentsSet02 Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 10.00)

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.4166667	3.0833335	0.9166666	4.0	45.0	1.1666667	0.4166667
Size	: 12					
Min Dist to Center	: 0.623610					
Max Dist to Center	: 4.801620					
Avg Dist to Center	: 1.754894					
Sum Squared Error	: 21.058727					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
2.0	2.0	0.0	4.0	45.0	1.0	1.0
1.0	3.0	2.0	4.0	45.0	2.0	0.0
1.0	3.0	2.0	4.0	45.0	1.0	0.0
1.0	3.0	1.0	4.0	45.0	1.0	1.0
1.0	3.0	1.0	4.0	45.0	1.0	0.0
1.0	2.0	1.0	4.0	45.0	1.0	0.0
1.0	2.0	1.0	4.0	45.0	1.0	0.0
1.0	2.0	1.0	4.0	45.0	1.0	1.0
1.0	2.0	1.0	4.0	45.0	1.0	1.0
1.0	2.0	1.0	4.0	45.0	1.0	1.0
1.0	7.0	0.0	4.0	45.0	1.0	0.0
5.0	6.0	0.0	4.0	45.0	2.0	0.0

Table 79: AccidentsSet02 Cluster 1 data from agglomerative clustering, euclidean distance centroid method threshold = 10.00)

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.59090908	3.54545454	0.5	5.4090909	35.0	1.18181819	0.227272727
Size	: 22					
Min Dist to Center	: 1.071802					
Max Dist to Center	: 6.513178					
Avg Dist to Center	: 3.074016					
Sum Squared Error	: 67.628343					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.0	4.0	0.0	4.0	35.0	3.0	1.0
3.0	4.0	0.0	4.0	35.0	2.0	0.0

Table 79: AccidentsSet02 Cluster 1 data from agglomerative clustering, euclidean distance centroid method threshold = 10.00)

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.59090908	3.54545454	0.5	5.4090909	35.0	1.18181819	0.227272727
Size	: 22					
Min Dist to Center	: 1.071802					
Max Dist to Center	: 6.513178					
Avg Dist to Center	: 3.074016					
Sum Squared Error	: 67.628343					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
3.0	4.0	0.0	4.0	35.0	1.0	1.0
2.0	4.0	0.0	4.0	35.0	1.0	1.0
2.0	6.0	0.0	4.0	35.0	1.0	0.0
2.0	6.0	0.0	3.0	35.0	1.0	1.0
1.0	4.0	0.0	2.0	35.0	1.0	0.0
1.0	4.0	0.0	4.0	35.0	1.0	0.0
1.0	5.0	1.0	3.0	35.0	1.0	0.0
1.0	2.0	1.0	9.0	35.0	1.0	0.0
1.0	2.0	1.0	9.0	35.0	1.0	0.0
1.0	2.0	1.0	9.0	35.0	1.0	0.0
2.0	1.0	0.0	9.0	35.0	1.0	0.0
1.0	1.0	0.0	9.0	35.0	1.0	0.0
1.0	1.0	0.0	6.0	35.0	1.0	0.0
3.0	3.0	0.0	6.0	35.0	1.0	0.0
2.0	2.0	0.0	6.0	35.0	1.0	0.0
2.0	3.0	0.0	6.0	35.0	1.0	0.0
1.0	2.0	1.0	7.0	35.0	1.0	0.0
1.0	3.0	1.0	6.0	35.0	1.0	0.0
1.0	6.0	5.0	3.0	35.0	2.0	0.0
2.0	9.0	0.0	2.0	35.0	1.0	1.0

Table 80: AccidentsSet02 Cluster 2 data from agglomerative clustering, euclidean distance centroid method threshold = 10.00)

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
4.0	25.0	0.0	2.0	70.0	2.0	0.0
Size	: 1					
Min Dist to Center	: 0.000000					
Max Dist to Center	: 0.000000					
Avg Dist to Center	: 0.000000					
Sum Squared Error	: 0.000000					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
4.0	25.0	0.0	2.0	70.0	2.0	0.0

Table 81: AccidentsSet02 Cluster 3 data from agglomerative clustering, euclidean distance centroid method threshold = 10.00)

Center	:					
VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
1.571428571	3.071428571	0.0	2.0	70.0	1.214285714	0.3571428571
Size	: 14					
Min Dist to Center	: 0.710705					
Max Dist to Center	: 7.450175					
Avg Dist to Center	: 2.655962					
Sum Squared Error	: 37.183466					

VE_TOTAL	PERSONS	PEDS	NO_LANES	SP_LIMIT	FATALS	DRUNK_DR
6.0	9.0	0.0	2.0	70.0	2.0	0.0
1.0	2.0	0.0	2.0	70.0	1.0	2.0
1.0	1.0	0.0	2.0	70.0	1.0	1.0
1.0	1.0	0.0	2.0	70.0	1.0	0.0
1.0	1.0	0.0	2.0	70.0	1.0	0.0
1.0	1.0	0.0	2.0	70.0	1.0	0.0
1.0	1.0	0.0	2.0	70.0	1.0	0.0
1.0	1.0	0.0	2.0	70.0	1.0	0.0
1.0	3.0	0.0	2.0	70.0	1.0	0.0
2.0	2.0	0.0	2.0	70.0	1.0	0.0
1.0	2.0	0.0	2.0	70.0	1.0	0.0
3.0	5.0	0.0	2.0	70.0	1.0	0.0
1.0	6.0	0.0	2.0	70.0	3.0	2.0
1.0	8.0	0.0	2.0	70.0	1.0	0.0

3.8.0.24 Observations Running k-means clustering with $k = 3$ resulted in 3 clusters split on the speed limit (SP_LIMIT) attribute. Increasing the number of clusters to $k = 5$ resulted in 5 clusters split on SP_LIMIT and number of lanes (NO_LANES) attributes, which greatly reduced the sum squared error of all the clusters. SP_LIMIT of 35.00 separated into 3 different clusters based on NO_LANES. Those clusters where Cluster 0 (Table 73) where NO_LANES was 9.0, Cluster 1 (Table 74) where NO_LANES was 6.0-7.0, and Cluster 3 (Table 76) where NO_LANES was 3.0-4.0. Cluster 2 (Table 75) contains all the 70.0 SP_LIMIT accidents on NO_LANES 2.0 roads. Cluster 2 has a deceptively high sum squared error and distance to the center due to an outlier accident involving 25 PERSONS when on average 4.5 PERSONS were involved in that cluster. Cluster 4 (Table 77) contains all the 45.0 SP_LIMIT and 4.0 NO_LANES accidents. In contrast the agglomerative clustering with euclidean distance, centroid method, and a threshold of 10.00 only had 4 clusters and did a more straightforward split on SP_LIMIT with the outlier data point with a high number of PERSONS being clusters on its own. Overall agglomerative clustered the data better than k-means for this data set as well.

3.9 AccidentsSet03

3.9.0.25 Best k-means clusters Initially $k = 2$ was used for AccidentsSet03.csv since it only contains 62 rows and there appeared to be a natural split in the data on NO_LANES attribute with values of 4.00 and 2.00 but otherwise the values within the attribute ranges did not have a lot of variation. After trying that, the variation in ranges of values in VE_TOTAL within those clusters became more apparent, and $k = 5$ appeared to result in the best data clustering, as shown in Tables 82, 83, 84, 85, and 86.

Table 82: AccidentsSet03Cluster 0 data from k-means clustering, euclidean distance $k = 5$

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.9285714285	4.0	1.1428571428	1.3571428571
Size	: 14			
Min Dist to Center	: 0.391230			
Max Dist to Center	: 2.647679			
Avg Dist to Center	: 0.696877			
Sum Squared Error	: 9.756274			

VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.0	4.0	1.0	1.0
1.0	0.0	4.0	3.0	3.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	2.0
1.0	1.0	4.0	1.0	2.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	2.0	4.0	1.0	2.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0

Table 83: AccidentsSet03Cluster 1 data from k-means clustering, euclidean distance $k = 5$

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.14285714285	2.0	1.1428571428	1.0
Size	: 14			
Min Dist to Center	: 0.202031			
Max Dist to Center	: 1.324803			
Avg Dist to Center	: 0.483584			
Sum Squared Error	: 6.770180			

VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.0	2.0	1.0	0.0
1.0	1.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0

Table 83: AccidentsSet03Cluster 1 data from k-means clustering, euclidean distance $k = 5$

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.14285714285	2.0	1.1428571428	1.0
Size	: 14			
Min Dist to Center	: 0.202031			
Max Dist to Center	: 1.324803			
Avg Dist to Center	: 0.483584			
Sum Squared Error	: 6.770180			

VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	2.0	2.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	2.0	1.0
1.0	1.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0

Table 84: AccidentsSet03Cluster 2 data from k-means clustering, euclidean distance $k = 5$

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
2.0909091	0.63636364	2.0	1.0	0.454545453
Size	: 11			
Min Dist to Center	: 0.661828			
Max Dist to Center	: 1.590260			
Avg Dist to Center	: 1.106557			
Sum Squared Error	: 12.172130			

VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
3.0	0.0	2.0	1.0	0.0
2.0	0.0	2.0	1.0	0.0
3.0	0.0	2.0	1.0	1.0
2.0	0.0	2.0	1.0	0.0
3.0	1.0	2.0	1.0	0.0
3.0	1.0	2.0	1.0	1.0
2.0	1.0	2.0	1.0	2.0
2.0	1.0	2.0	1.0	1.0
1.0	1.0	2.0	1.0	0.0
1.0	1.0	2.0	1.0	0.0
1.0	1.0	2.0	1.0	0.0

Table 85: AccidentsSet03Cluster 3 data from k-means clustering, euclidean distance $k = 5$

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.09090908	0.81818182	4.0	1.18181819	0.0
Size	: 11			
Min Dist to Center	: 0.272727			
Max Dist to Center	: 1.471492			
Avg Dist to Center	: 0.462426			
Sum Squared Error	: 5.086687			

VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.0	4.0	2.0	0.0
2.0	0.0	4.0	2.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0

Table 86: AccidentsSet03Cluster 4 data from k-means clustering, euclidean distance $k = 5$

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
4.1666667	0.0	3.66666665	1.0	0.5
Size	: 12			
Min Dist to Center	: 0.623610			
Max Dist to Center	: 5.864204			
Avg Dist to Center	: 1.710802			
Sum Squared Error	: 20.529627			

VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
5.0	0.0	4.0	1.0	1.0
5.0	0.0	4.0	1.0	0.0
4.0	0.0	4.0	1.0	0.0
10.0	0.0	4.0	1.0	1.0
4.0	0.0	2.0	1.0	0.0
4.0	0.0	2.0	1.0	0.0
3.0	0.0	4.0	1.0	1.0
3.0	0.0	4.0	1.0	0.0
3.0	0.0	4.0	1.0	0.0
3.0	0.0	4.0	1.0	1.0
3.0	0.0	4.0	1.0	0.0
3.0	0.0	4.0	1.0	2.0

3.9.0.26 Best Agglomerative clusters Agglomerative clustering was initially run with euclidean distance, single link cluster distance, and no threshold to view the full tree structure and determine a good threshold for assigning clusters. It was found that using the centroid method with euclidean distance and a threshold of 2.00 produced the 6 clusters with the best fit to the data by separating on NO_LANES and VE_TOTAL primiarly while separating an outlier its own cluster. These clusters are shown in Tables 87, 88, 89, 90, 91, and 92.

Table 87: AccidentsSet03 Cluster 0 data from agglomerative clustering, euclidean distance centroid method threshold = 2.00)

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
10.0	0.0	4.0	1.0	1.0
Size	: 1			
Min Dist to Center	: 0.000000			
Max Dist to Center	: 0.000000			
Avg Dist to Center	: 0.000000			
Sum Squared Error	: 0.000000			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
10.0	0.0	4.0	1.0	1.0

Table 88: AccidentsSet03 Cluster 1 data from agglomerative clustering, euclidean distance centroid method threshold = 2.00)

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.0	4.0	3.0	3.0
Size	: 1			
Min Dist to Center	: 0.000000			
Max Dist to Center	: 0.000000			
Avg Dist to Center	: 0.000000			
Sum Squared Error	: 0.000000			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.0	4.0	3.0	3.0

Table 89: AccidentsSet03 Cluster 2 data from agglomerative clustering, euclidean distance centroid method threshold = 2.00)

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
3.33333335	0.33333333	2.0	1.0	0.33333333
Size	: 6			
Min Dist to Center	: 0.577350			
Max Dist to Center	: 1.000000			
Avg Dist to Center	: 0.807223			
Sum Squared Error	: 4.843337			

VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
4.0	0.0	2.0	1.0	0.0
4.0	0.0	2.0	1.0	0.0
3.0	0.0	2.0	1.0	0.0
3.0	0.0	2.0	1.0	1.0
3.0	1.0	2.0	1.0	0.0
3.0	1.0	2.0	1.0	1.0

Table 90: AccidentsSet03 Cluster 3 data from agglomerative clustering, euclidean distance centroid method threshold = 2.00)

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
3.55555554	0.0	4.0	1.0	0.55555556
Size	: 9			
Min Dist to Center	: 0.711458			
Max Dist to Center	: 1.547599			
Avg Dist to Center	: 1.010874			
Sum Squared Error	: 9.097869			

VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
5.0	0.0	4.0	1.0	1.0
5.0	0.0	4.0	1.0	0.0
4.0	0.0	4.0	1.0	0.0
3.0	0.0	4.0	1.0	0.0
3.0	0.0	4.0	1.0	0.0
3.0	0.0	4.0	1.0	0.0
3.0	0.0	4.0	1.0	2.0
3.0	0.0	4.0	1.0	1.0
3.0	0.0	4.0	1.0	1.0

Table 91: AccidentsSet03 Cluster 4 data from agglomerative clustering, euclidean distance centroid method threshold = 2.00)

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.1904761905	0.3333333333	2.0	1.0953	0.8095238095
Size	: 21			
Min Dist to Center	: 0.439026			
Max Dist to Center	: 1.589364			
Avg Dist to Center	: 0.814757			
Sum Squared Error	: 17.109890			

VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
2.0	1.0	2.0	1.0	2.0
2.0	1.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	0.0
2.0	0.0	2.0	1.0	0.0
2.0	0.0	2.0	1.0	0.0
1.0	1.0	2.0	1.0	1.0
1.0	1.0	2.0	1.0	1.0
1.0	1.0	2.0	1.0	0.0
1.0	1.0	2.0	1.0	0.0
1.0	1.0	2.0	1.0	0.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	1.0	1.0
1.0	0.0	2.0	2.0	2.0
1.0	0.0	2.0	2.0	1.0

Table 92: AccidentsSet03 Cluster 5 data from agglomerative clustering, euclidean distance centroid method threshold = 2.00)

Center	:			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.041666667	0.916666666	4.0	1.083333333	0.666666666
Size	: 24			
Min Dist to Center	: 0.356000			
Max Dist to Center	: 1.744535			
Avg Dist to Center	: 0.745421			
Sum Squared Error	: 17.890104			
VE_TOTAL	PEDS	NO_LANES	FATALS	DRUNK_DR
1.0	0.0	4.0	2.0	0.0
2.0	0.0	4.0	2.0	0.0
1.0	2.0	4.0	1.0	2.0
1.0	1.0	4.0	1.0	2.0
1.0	1.0	4.0	1.0	2.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	1.0	4.0	1.0	0.0
1.0	0.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0
1.0	1.0	4.0	1.0	1.0

3.9.0.27 Observations

4 Analysis

Both k-means and agglomerative clustering managed to create interesting clusters, many of which made sense.

A notable problem with k-means is the necessity to force a k to cluster the data into. For too large a k, one can end up with empty clusters. For too small, the clusters are meaningless. We came across these issue time and time again. It became a balancing act of trying to get the right cluster metrics amongst all of the k clusters made — we didn't want some to be too big and others too small and so on. When we could plot the data, it was easy to see how many clusters one might pick, so choosing a k for those data was straightforward. For others, it was, understandably, more difficult.

Agglomerative, on the other hand, did a fantastic job of clustering the data. Using the algorithm was easier — where we might plot the data with k-means, with agglomerative we could see the structure via the dendrogram before trimming. This gave us a nice intuition for "where" to "trim" the dendrogram into clusters. Though it wasn't investigated in this paper, using different distance metrics — datapoint and cluster — changed the dendrogram's shape in interesting ways that were interesting to think about. With this dendrogram, data fell nicely into clusters that made sense and if the cluster felt too general, we could be more discriminate with our trimming threshold and not worry that any random seeding (which is present in k-means) would create artifacts unique to a particular run of the algorithm.

Outlier detection between the two were starkly different. Agglomerative seemed to have outlier detection built in. If a data point or two are too far away, they become their own singleton cluster, which we could ignore or consider on its own.

Overall, we would choose to go forward with the agglomerative clustering. The open ended approach make for much more interesting data discovery than with the k-means. With k-means, it felt too much like we were trying to fit our algorithm to what we already knew with the data, which became frustrating and seems counterintuitive to the discovery process.

Appendix A

A.1 Code Snippets

```
def disk_k_means(self, D, k):
   assert k < D.size(), 'k(%d) is larger than data(%d)' % (k, D.size())
   clusters = None
   means = self.select_initial_clusters(D, k)
   repeat = True
   while self.stopping_criteria(k, means, clusters) == True:
      print('calculating')
      family = [D.dimensions()*[0] for j in range(k)]# family of vectors of size dim(D)
      num_points = [0 for j in range(k)]             # number of points in each cluster
      clusters = [[] for j in range(k)]              # actual clusters
      for x in D:
         j = self.arg_min(k, x, means)
         clusters[j].append(x)
         family[j] = [a + b for a,b in zip(family[j], x)]
         num_points[j] += 1
      for j in range(k):
         if num_points[j]:
            means[j] = [s / num_points[j] for s in family[j]]
   return means, clusters
```

Figure 1: Disk k-means algorithm implemented in Python

```
def get_all_clusters(tree):
   clusters = []
   get_all_clusters_rec(clusters, tree)
   return clusters
def get_all_clusters_rec(clusters, element):
   if element.tag == 'leaf':
      clusters.append(element.attrib['data'])
   for e in element:
      get_all_clusters_rec(clusters, e)

def get_branches_rec(branches, element, threshold):
   for e in element:
      if e.tag == 'tree' or e.tag == 'node' or e.tag == 'trimmed_tree':
         if float(e.attrib['height']) < threshold:
            branches.append(e)
         else:
            get_branches_rec(branches, e, threshold)
      elif e.tag == 'leaf':
         branches.append(e)

def get_clusters(tree, threshold):
   centroids = None
   clusters  = []
   branches = []
   trimmed_tree = ElementTree.Element('trimmed_tree')
   trimmed_tree.set('threshold', '%.3f' % threshold)

   get_branches_rec(branches, tree, threshold)
   for b in branches:
      trimmed_tree.append(b)
   for stem in branches:
      c = get_all_clusters(stem)
      print_tree(stem)
      print(c)
      clusters.append(c)
   print(clusters)
   centroids = calc_centroids(clusters)

   return trimmed_tree, centroids, clusters
```

Figure 2: Dendroid tree trimming strategy