DSC 520 Week 7 Clustering

Demond Love

7/21/2018

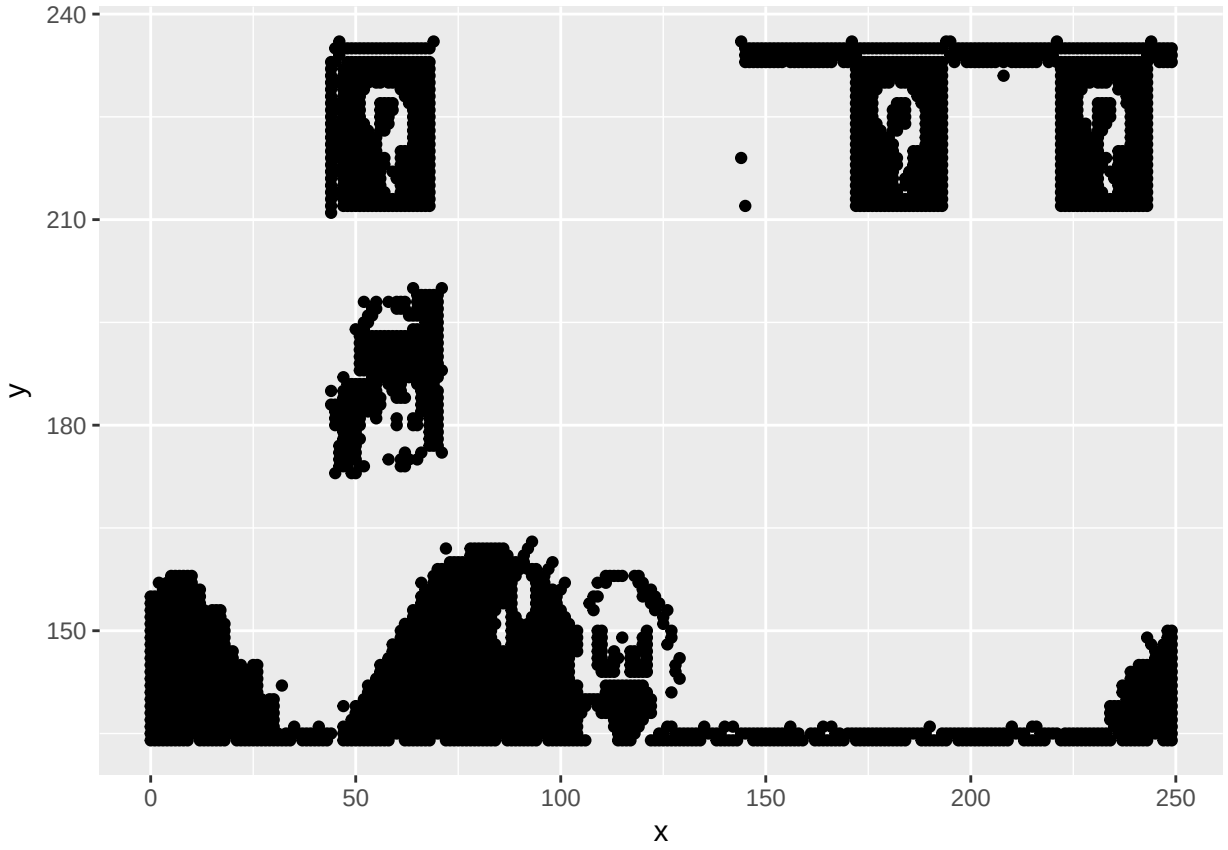
Labeled data is not always available. For these types of datasets, you can use unsupervised algorithms to extract structure. The k-means clustering algorithm and the k nearest neighbor algorithm both use the Euclidean distance between points to group data points. The difference is the k-means clustering algorithm does not use labeled data.

In this problem, you will use the k-means clustering algorithm to look for patterns in an unlabeled dataset. The dataset for this problem is found at data/clustering-data.csv.

```
library(ggplot2)
setwd('/Users/Love/Documents/DSC 520 Statistics for Data Science/Week 7 DSC 520')
file = read.csv('./clustering-data.csv')
```

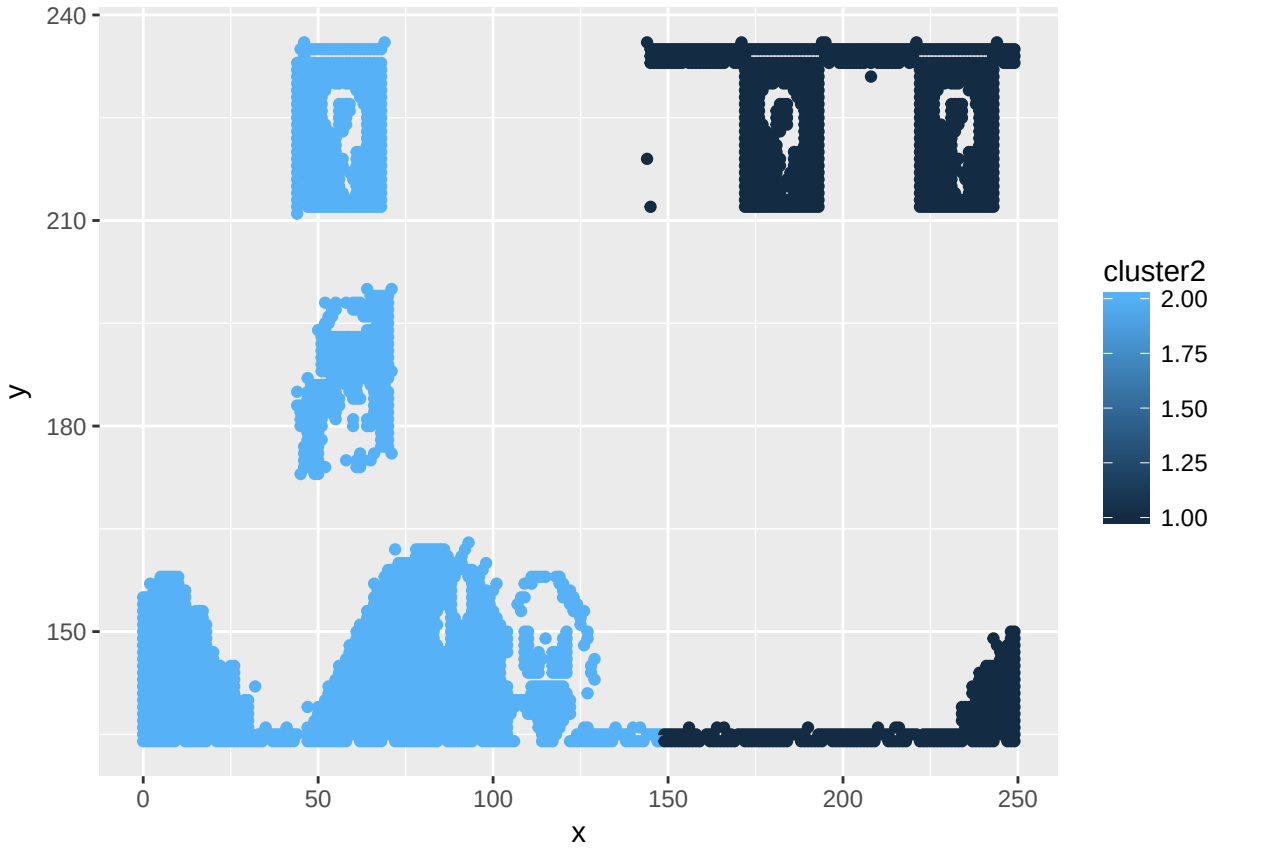
a. Plot the dataset using a scatter plot.

```
ggplot(file, aes(x, y)) + geom_point()
```

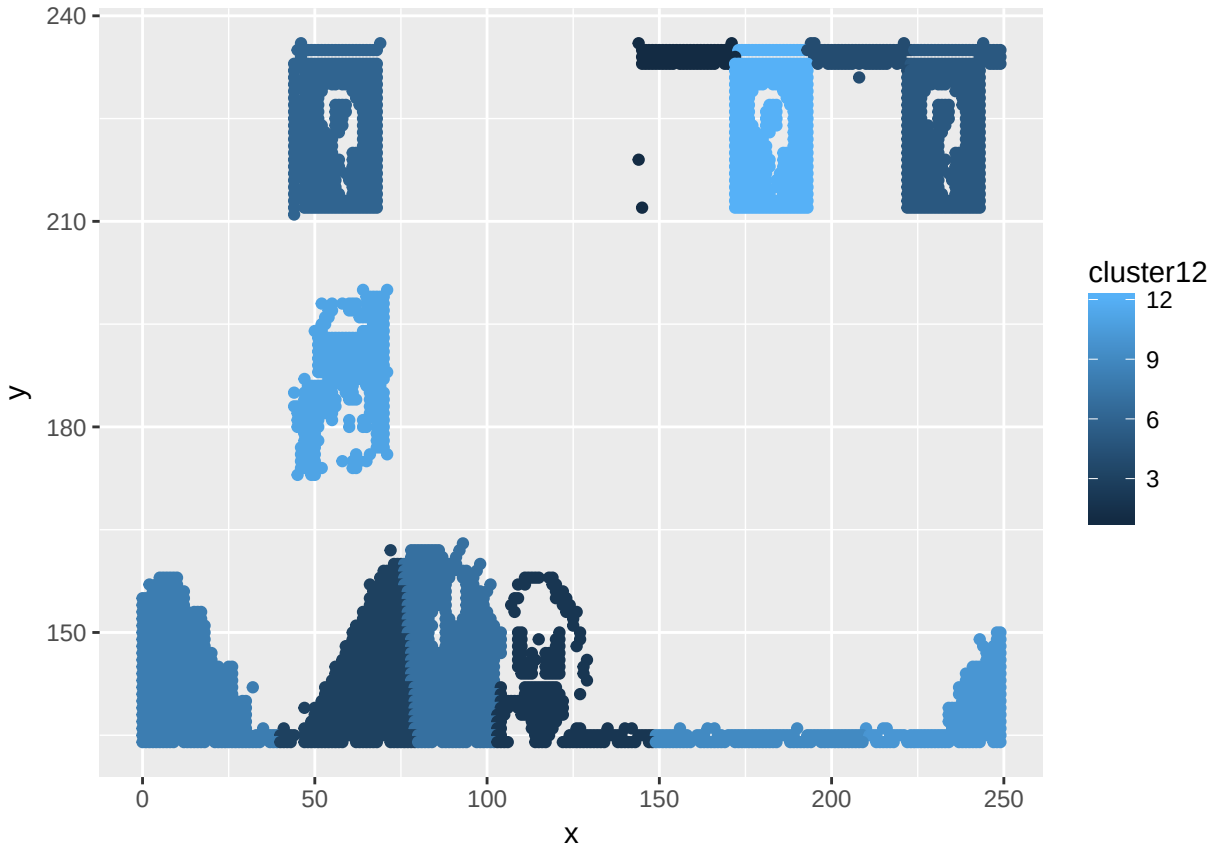


b. Fit the dataset using the k-means algorithm from k=2 to k=12. Create a scatter plot of the resultant clusters for each value of k.

```
cl2 = kmeans(file, centers = 2)
file$cluster2 = cl2$cluster
ggplot(file, aes(x, y)) + geom_point(aes(color = cluster2))
```




```
cl12 = kmeans(file, centers = 12)
file$cluster12 = cl12$cluster
ggplot(file, aes(x, y)) + geom_point(aes(color = cluster12))
```



c. As k-means is an unsupervised algorithm, you cannot compute the accuracy as there are no correct values to compare the output to. Instead, you will use the average distance from the center of each cluster as a measure of how well the model fits the data. To calculate this metric, simply compute the distance of each data point to the center of the cluster it is assigned to and take the average value of all of those distances.

This value is essentially the total within sum of square, which is defined as the summation of all the clusters over the sum of squared Euclidean distances between items and their corresponding centroid. The kmeans function calculates this for us, so if we can simply take the average value from here.

Calculate this average distance from the center of each cluster for each value of k and plot it as a line chart where k is the x-axis and the average distance is the y-axis.

```
str(cl2)
```

```
## List of 9
##  $ cluster     : int [1:4022] 2 2 1 1 1 1 1 1 2 2 ...
##  $ centers     : num [1:2, 1:2] 207.8 62.3 203.3 162.4
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:2] "x" "y"
##  $ totss       : num 28608985
##  $ withinss    : num [1:2] 3057964 5385717
##  $ tot.withinss: num 8443681
##  $ betweenss   : num 20165304
```

```
##  $ size        : int [1:2] 1308 2714
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

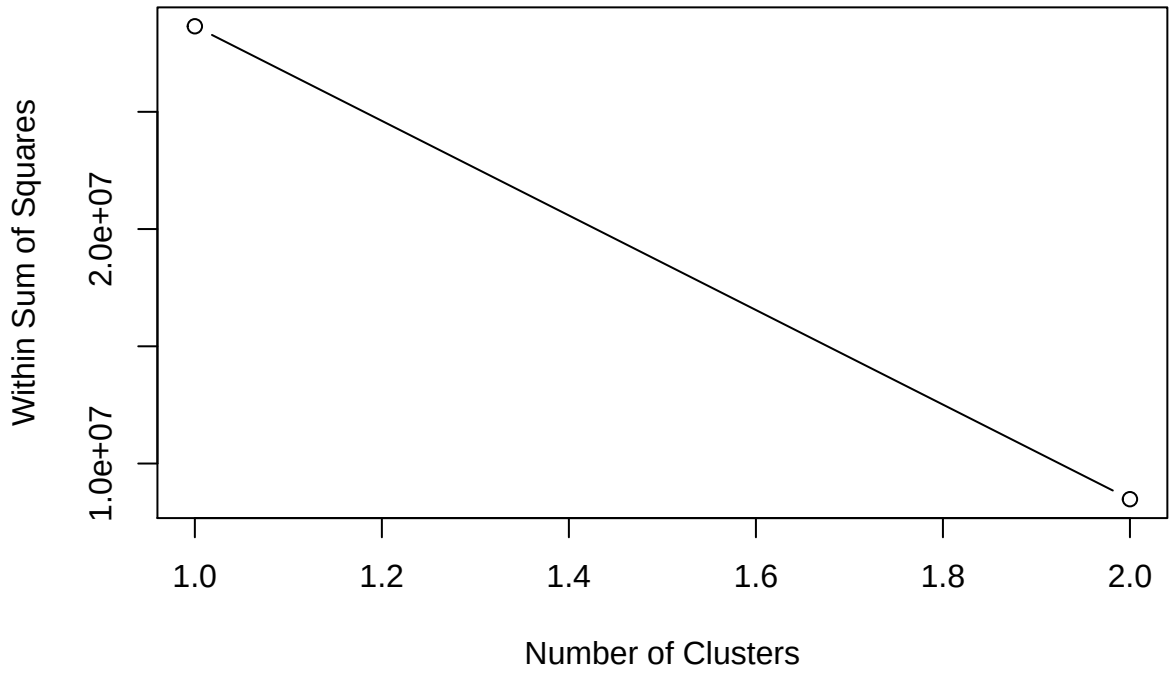
```
cl2$withinss/cl2$size
```

```
## [1] 2337.893 1984.420
```

```
wss = (nrow(file)-1)*sum(apply(file,2,var))

for (i in 1:2) wss[i] = sum(kmeans(file, centers = i)$withinss)

plot(1:2, wss, type = 'b', xlab="Number of Clusters", ylab = "Within Sum of Squares")
```



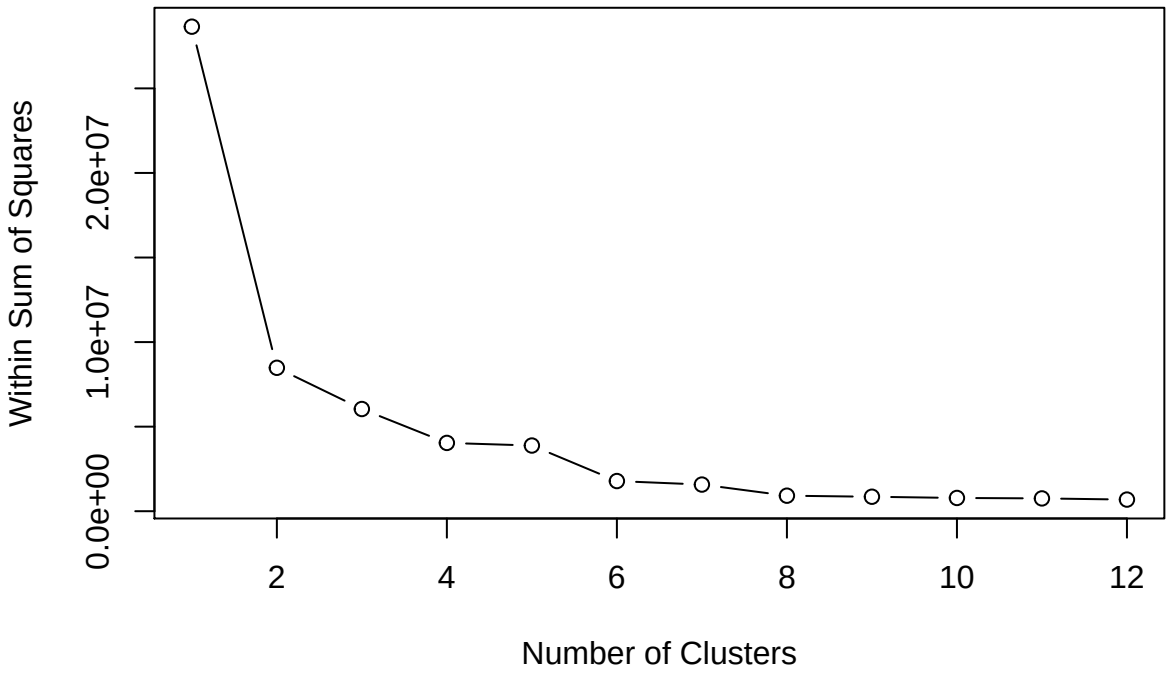

```
cl12$withinss/cl12$size
```

```
##  [1]  74.79348 143.81635 116.29340  67.46675 105.47531 107.35933 119.30163
##  [8] 120.34213 326.60721 104.52452 104.91110  97.30292
```

```
wss = (nrow(file)-1)*sum(apply(file,2,var))

for (i in 2:12) wss[i] = sum(kmeans(file, centers = i)$withinss)

plot(1:12, wss, type = 'b', xlab="Number of Clusters", ylab = "Within Sum of Squares")
```



d. One way of determining the "right" number of clusters is to look at the graph of k versus average distance and finding the "elbow point". Looking at the graph you generated in the previous example, what is the elbow point for this dataset?

I would say that the elbow point for this dataset is at 5 clusters.