K-Means Clustering

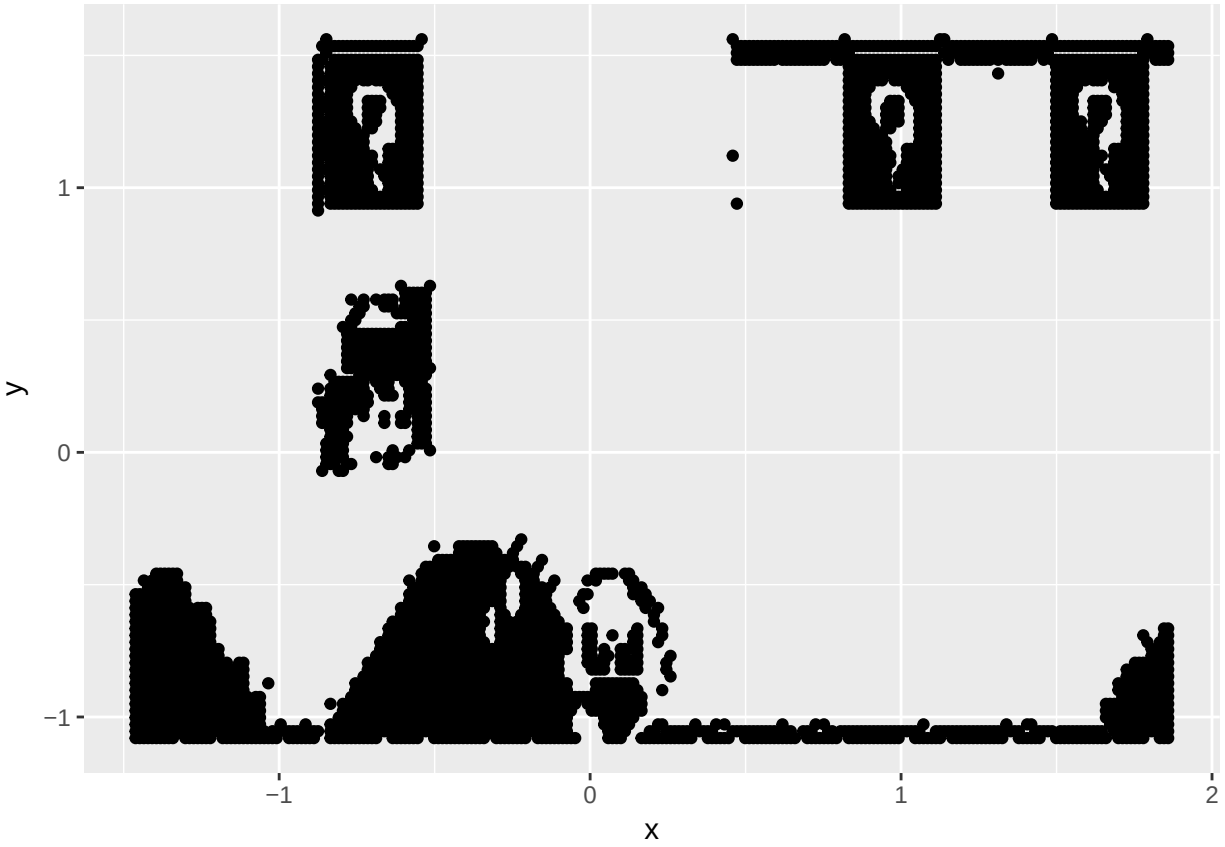
Demond Love

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.

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

Ploting the dataset using a scatter plot and reviewing the data statistically.

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




```
summary(file_z)
```

```
##        x                  y
##  Min.   :-1.4617   Min.   :-1.0797
##  1st Qu.:-0.7149   1st Qu.:-0.8985
##  Median :-0.3681   Median :-0.5619
```

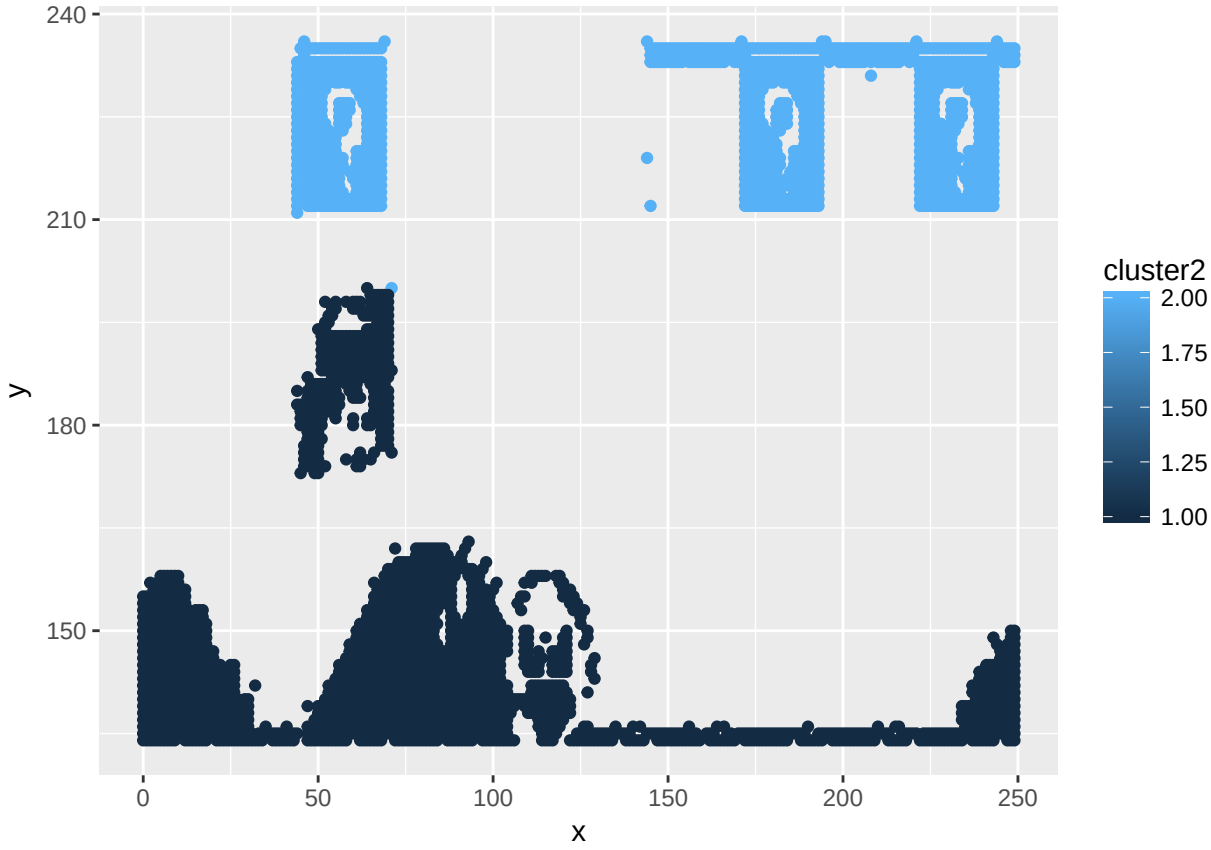
```
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.9387   3rd Qu.: 1.0951
##  Max.   : 1.8589   Max.   : 1.5611
```

```
str(file_z)
```

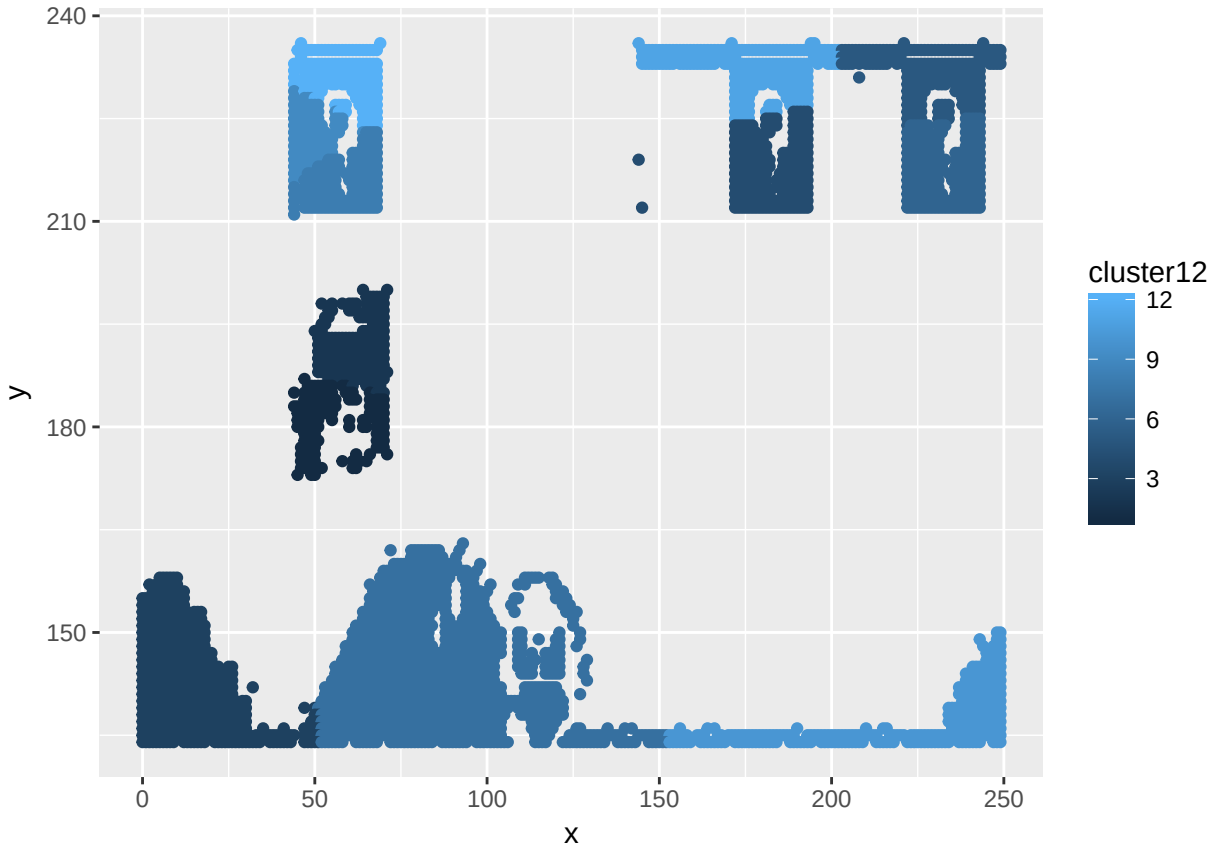
```
## 'data.frame':    4022 obs. of  2 variables:
##  $ x: num  -0.848 -0.542 0.459 0.819 1.125 ...
##  $ y: num  1.56 1.56 1.56 1.56 1.56 ...
```

Fitting the dataset using the k-means algorithm from k=2 to k=12 and creating a scatter plot of the resulting clusters for each value of k.

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




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



Calculating the average distance from the center of each cluster for each value of k and plotting them as a line chart where k is the x-axis and the average distance is the y-axis.

```
str(cl2)
```

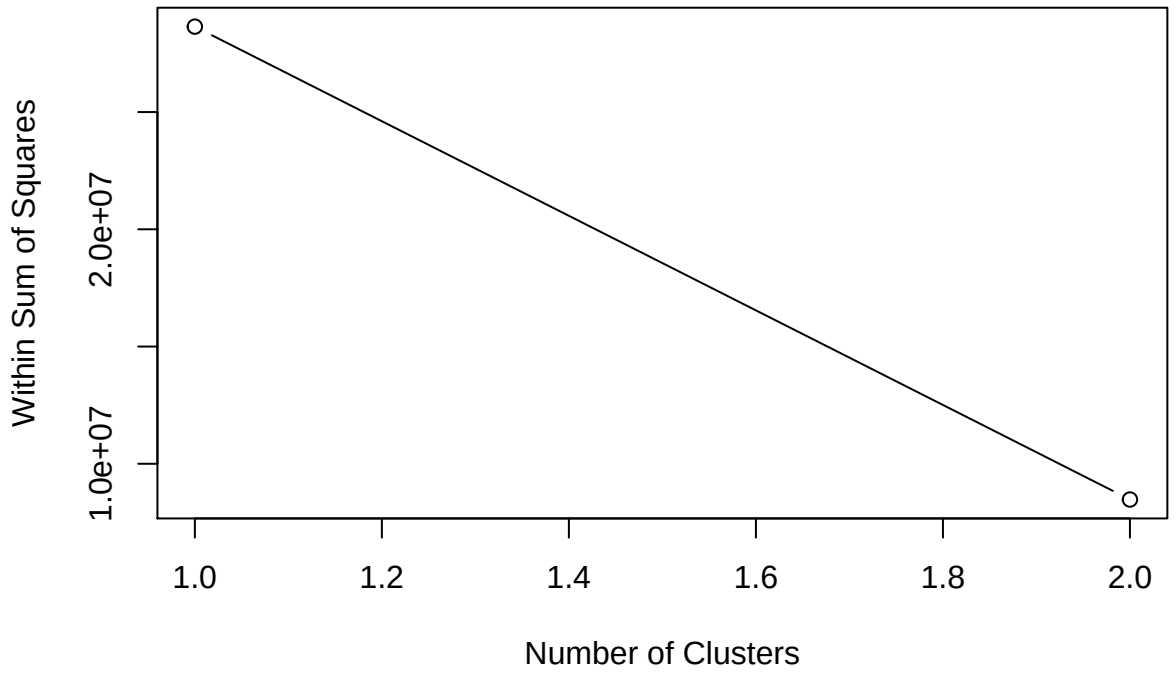
```
## List of 9
##  $ cluster     : int [1:4022] 2 2 2 2 2 2 2 2 2 2 ...
##  $ centers     : num [1:2, 1:2] -0.357 0.649 -0.691 1.257
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:2] "x" "y"
##  $ totss       : num 8042
##  $ withinss    : num [1:2] 2235 1380
##  $ tot.withinss: num 3615
##  $ betweenss   : num 4427
##  $ size        : int [1:2] 2595 1427
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

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

```
## [1] 0.8614348 0.9669392
```

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

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

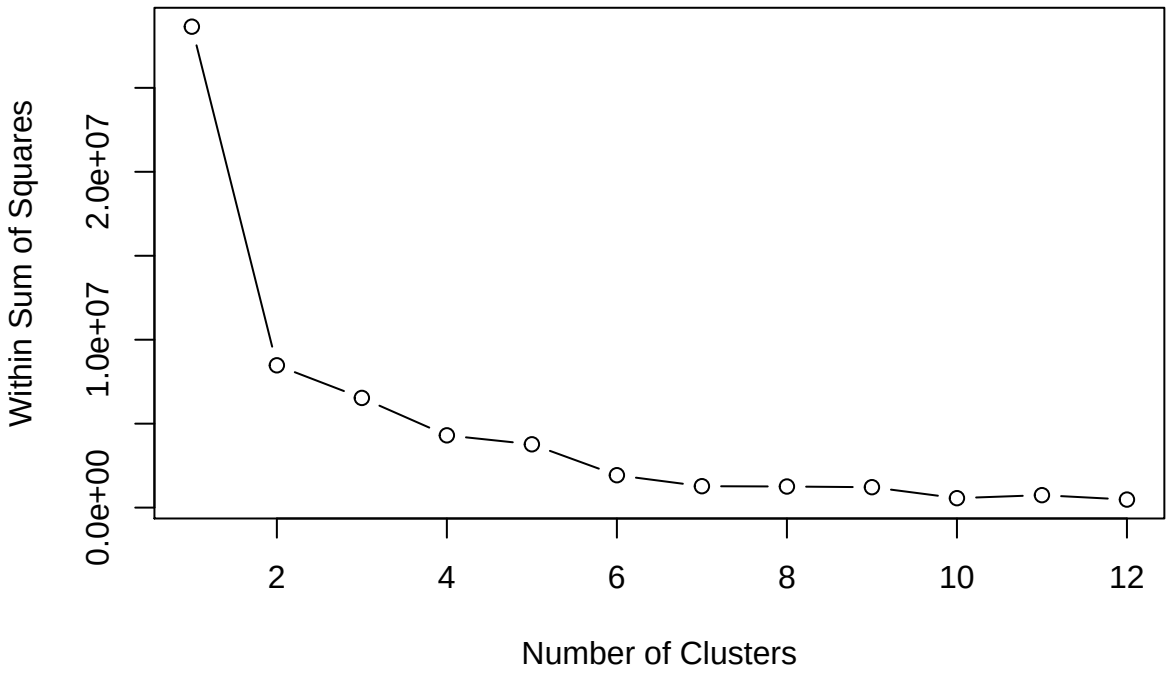
```
plot(1:2, wss, type = 'b', xlab="Number of Clusters", ylab = "Within Sum of Squares")
```




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

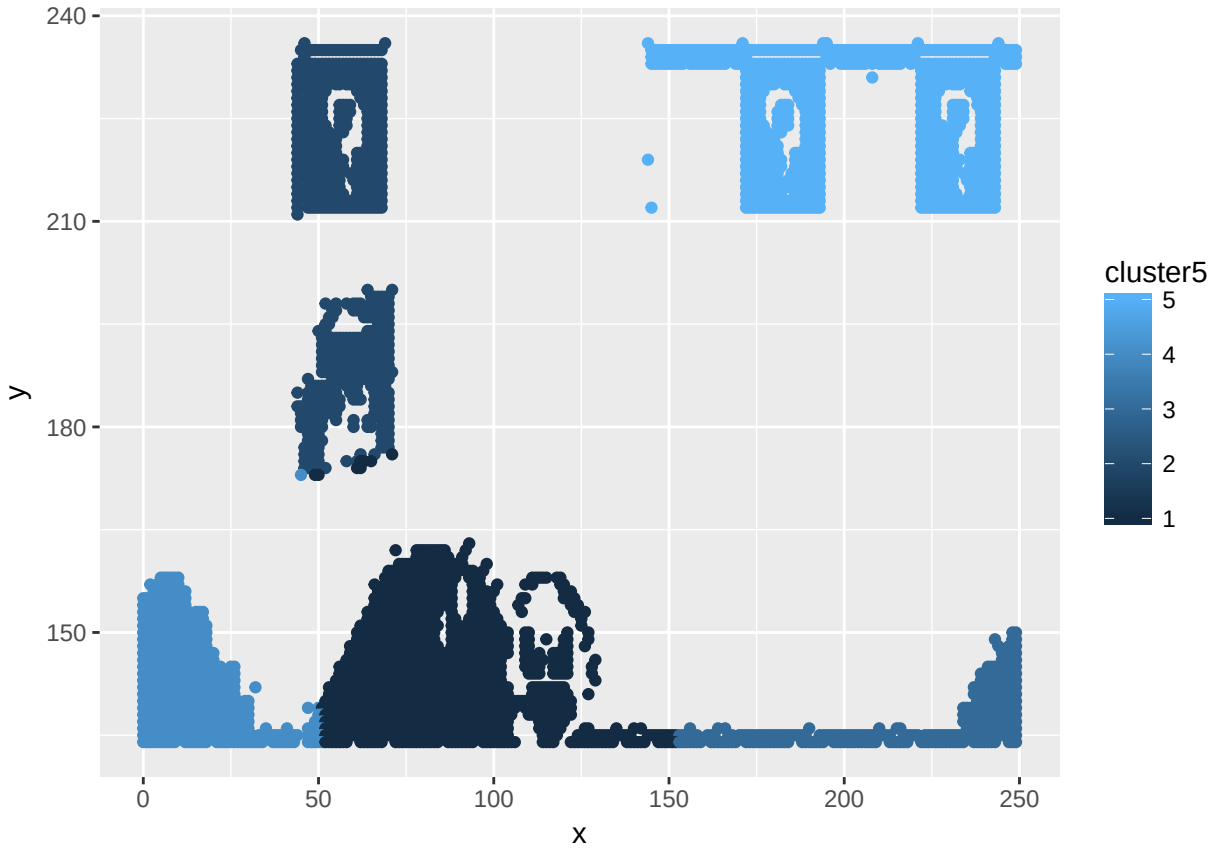
```
##  [1] 0.021763496 0.015357828 0.053487952 0.021252051 0.030549314
##  [6] 0.018031143 0.110875889 0.014566839 0.008800706 0.164618263
## [11] 0.043893180 0.015048711
```

```
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". The elbow point for this dataset is 5.

```
cl5 = kmeans(file_z, centers = 5)
file$cluster5 = cl5$cluster
ggplot(file, aes(x, y)) + geom_point(aes(color = cluster5))
```




```
str(cl5)
```

```
## List of 9
##  $ cluster     : int [1:4022] 2 2 5 5 5 5 5 5 2 2 ...
##  $ centers     : num [1:5, 1:2] -0.309 -0.696 1.488 -1.269 1.259 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:5] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:2] "x" "y"
##  $ totss       : num 8042
##  $ withinss    : num [1:5] 155.3 194.1 51.9 32.4 177
##  $ tot.withinss: num 611
##  $ betweenss   : num 7431
##  $ size        : int [1:5] 1357 773 315 592 985
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```