15 Minutes Lecture Challenge

Milad Korde

One way to have a more appropriate understanding of data is to find points showing similar values. One way to satisfy this criterion is to use the clustering method, an algorithm to group observations based on similarity rate. The technique we are going to use is considered an *agglomerative approach* which, in simple words, means stitching data points together step by step until every point is associated with a certain class.

Before that, we need to add a field (column) to our sp1 data point with random numbers created by the *sample* function. The numerical values in the *Rvalue* field will be used to run the cluster analysis:

```
sp1$RValue<- c(sample(1:139))
```

Now let us check the data frame:

```
head(sp1@data)
  X             taxon latitude longitude type RValue
1 1 Cucurbita_cordata 28.95470 -113.5625    G     36
2 2 Cucurbita_cordata 24.28577 -111.0110    H     81
3 3 Cucurbita_cordata 25.93470 -111.5421    H    118
4 4 Cucurbita_cordata 25.94749 -111.7300    H      9
5 5 Cucurbita_cordata 25.95765 -111.4609    H     68
6 6 Cucurbita_cordata 26.20671 -111.6175    H     97
```

Now we have some numerical values to use in our cluster analysis. One important thing to take into consideration is the concept of closeness in clustering. The method we use will be based on *Euclidean distance*, where a straight line in a vector system will define how close data points are to each other.

Before moving to the cluster, we need to create a vector:

```
d3 <- sp1[]
### Generate a Spatial Point Dataframe
names(d3)
plot (d3)
```

Now we prepare the data and tools for K-means:

The animation package will help us create a step-by-step clustering system that is visually more appealing than using a single function to create the clusters.

```
install.packages("animation", dependencies = TRUE)
```

Also, we add *dplyr* library to help us take a look at the data. You may get some warnings for the version, but the library will work the same.

```
library(dplyr) # to use glimpse function, which provides another way to look at the data
```

Now we convert our data frame to CSV. Converting the data frame to a CSV will make the process available for non-geospatial data as well:

```
write.csv(d3,"E:/Research-and-Program-Coordinator-main/Research-and-Program-Coordinator-main/data/d3.csv", row.names = TRUE)

PATH <-"E:/Research-and-Program-Coordinator-main/Research-and-Program-Coordinator-main/data/d3.csv"

df <- read.csv(PATH)

#Check the result
glimpse(df)
```

While there are many clustering algorithms, the one that we use is called *K-means*, in which a centroid for each cluster will be constructed first, and then distances will be calculated between the actual value of a point and the centroid. This process will repeat itself up to the point where no data point is moved from one cluster to another. While there are several techniques to determine the number of clusters, such as the *Elbow method* and *Pseudo F-statistic*, we go with trial and error since the number of observations is not huge. We can always look at the result, use different numbers, and make sure that the number of clusters is fine.

We can also use the *animation* library to provide a better visualization process for our clustering process:

```
library(animation) # You may get some warnings about the R version
```

Now we can run the cluster analysis:

```
#Run the K-means
kmeans.ani(df, 3)
```

Now the interesting part starts. By running the following code, you will be shown the process of cluster creation in your viewer section:

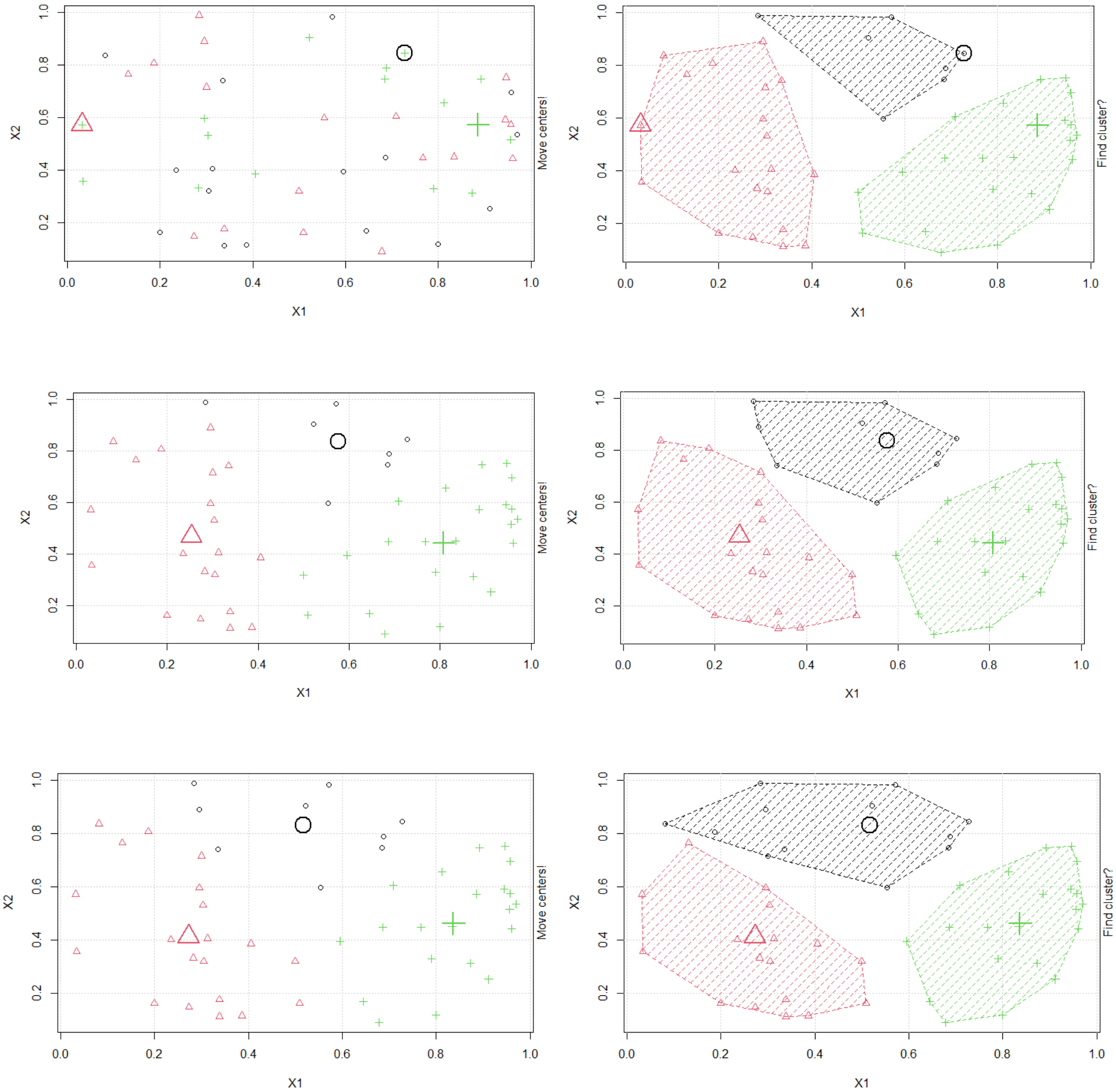
```
plot (kmeans.ani())
```

What happened here consists of calculating a centroid and eventually adjusting the distances between data points and the centroid to create an “*among cluster difference and between data point similarity*.” We ended up having three clusters where the values are close to each other, and they can be considered similar.

15 Minutes Lecture Challenge

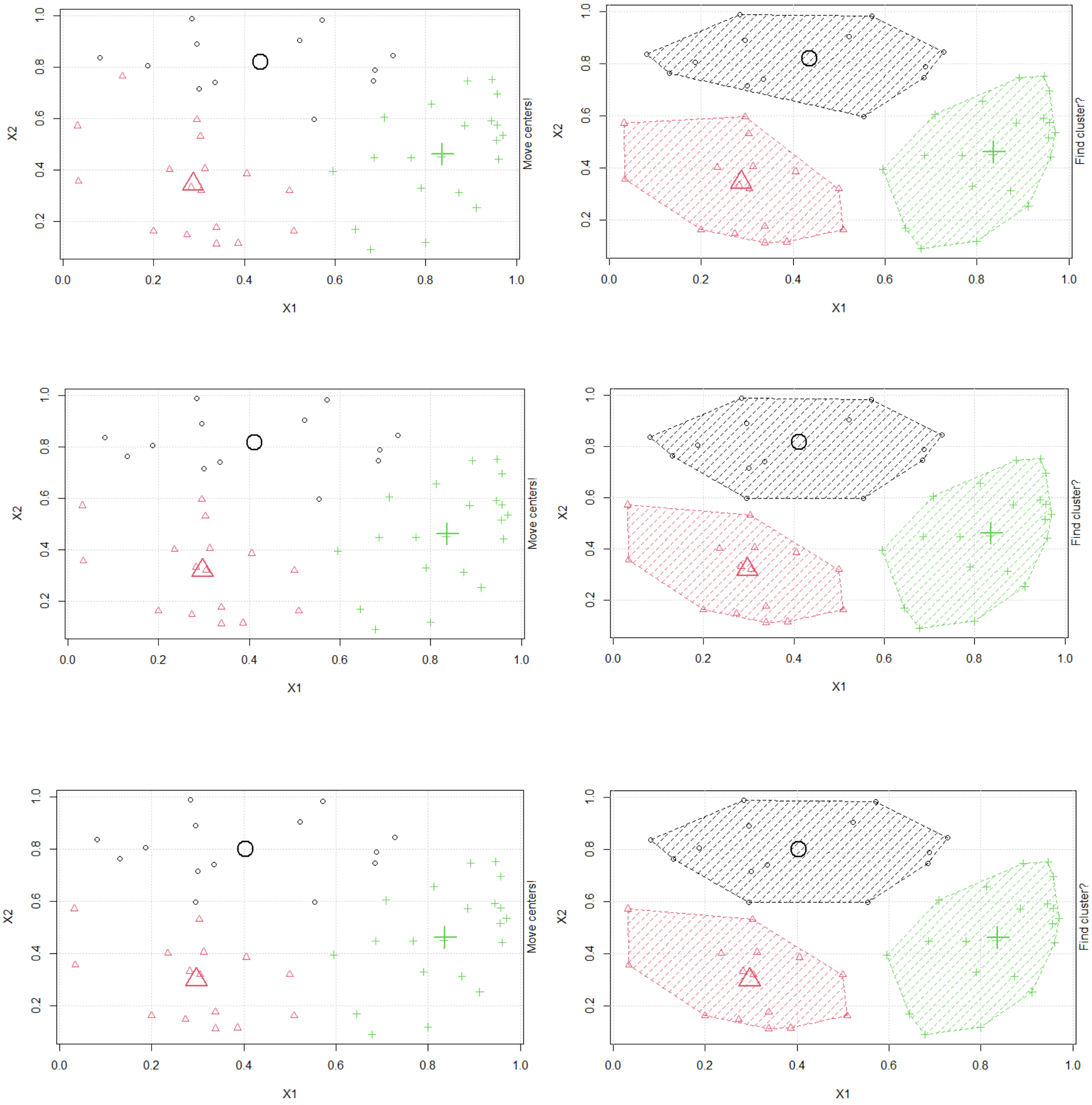
Milad Korde

The following figures show the process and centroid movement after each iteration. The figures on the left side show the centroid movement process, while the right column shows the found clusters. The last figure on the right column is the final result where there is no need to move the centroids further.



15 Minutes Lecture Challenge

Milad Korde


X1
X2
Move centers!
Find cluster?
Move centers!
Find cluster?
Move centers!
Find cluster?