Clustering

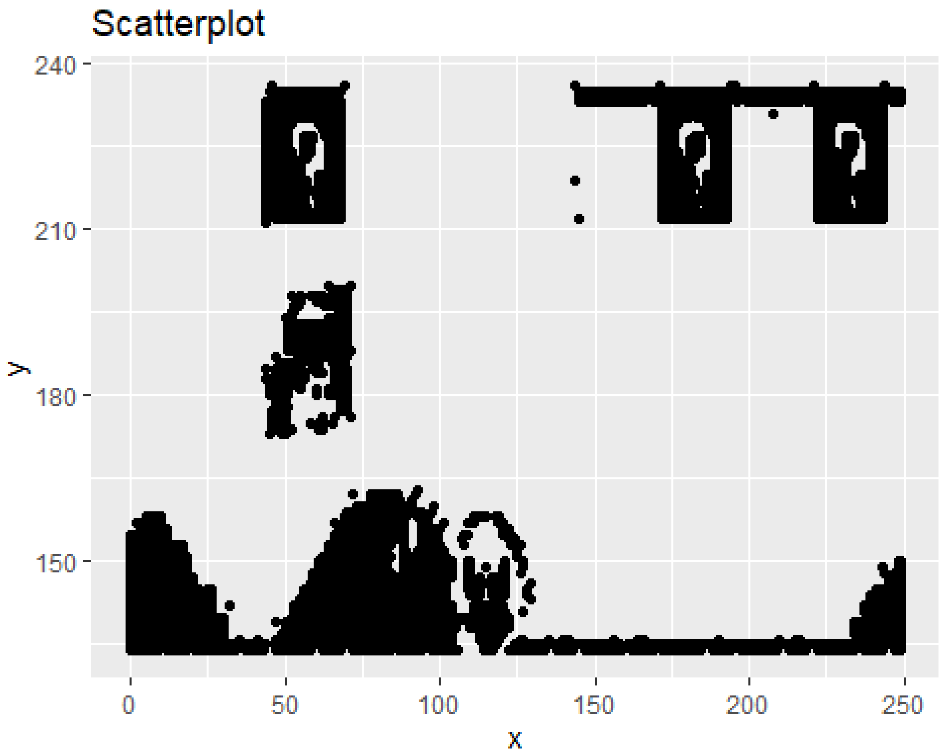
Marc Riley

6/2/2021

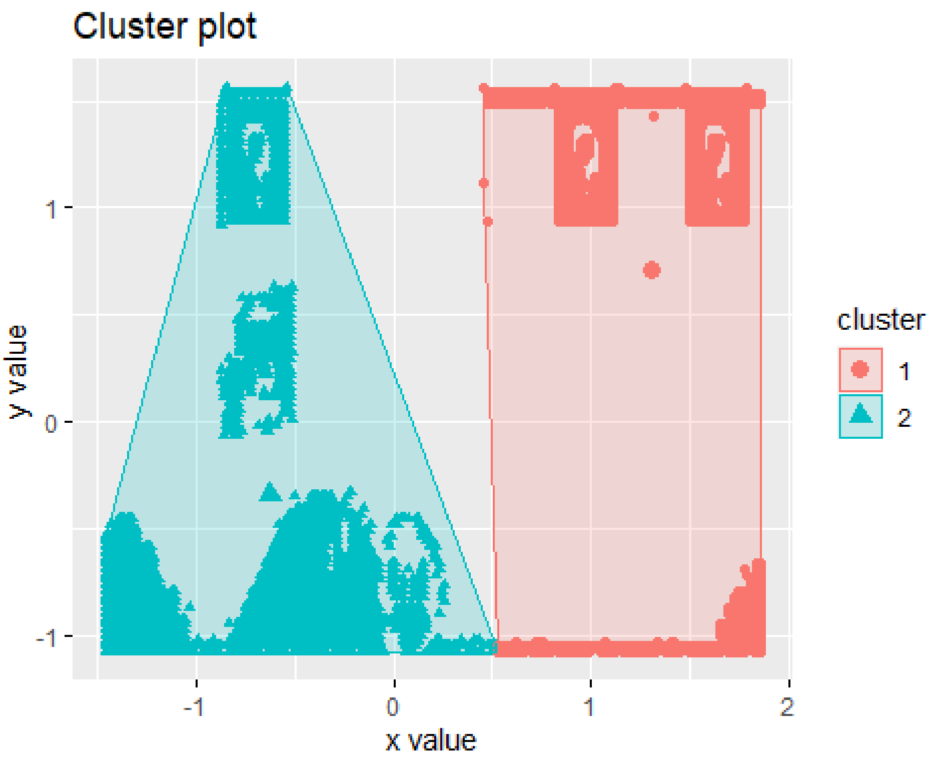
```
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(cluster)
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.0.5

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

library(ggplot2)

cluster_df <- read.csv("clustering-data.csv")

# Look at the data
head(cluster_df)
```

```
##      x   y
## 1  46 236
## 2  69 236
## 3 144 236
## 4 171 236
## 5 194 236
## 6 195 236
```

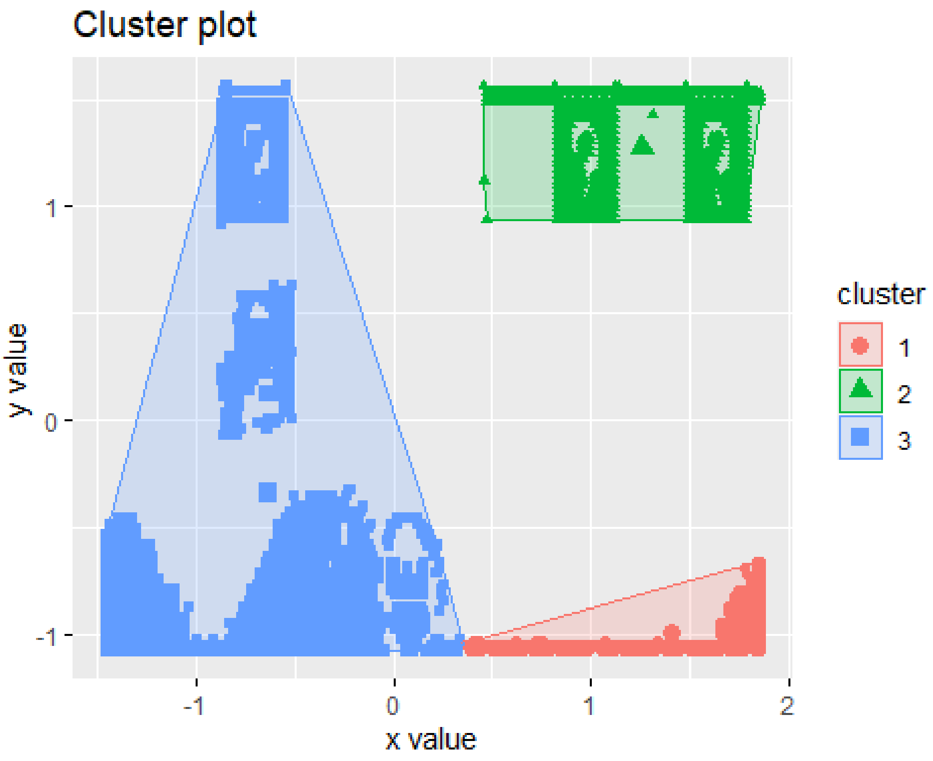
```
# plot the data

ggplot(data =  cluster_df, aes(x=x, y=y)) +
  geom_point() +
  ggtitle("Scatterplot")
```



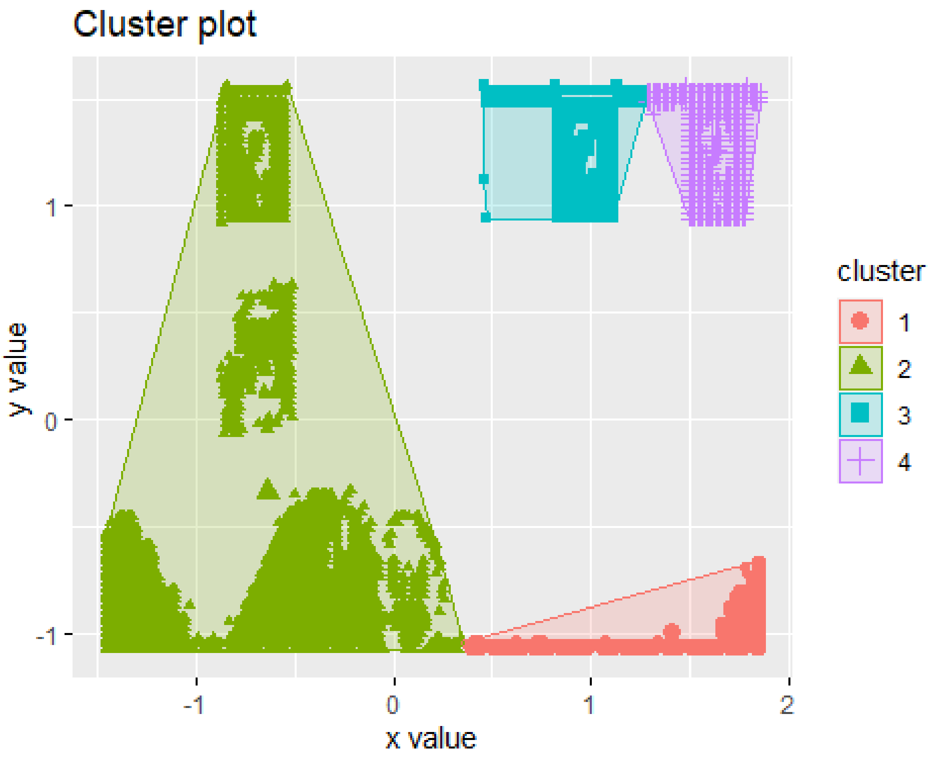
```
# Fit the dataset using the k-means algorithm from k=2 to k=12. Create a
scatter plot of the resultant clusters for each value of k.

#k-means2
kmean2 <- kmeans(cluster_df, 2,)
fviz_cluster(kmean2, geom = "point", data = cluster_df)
```



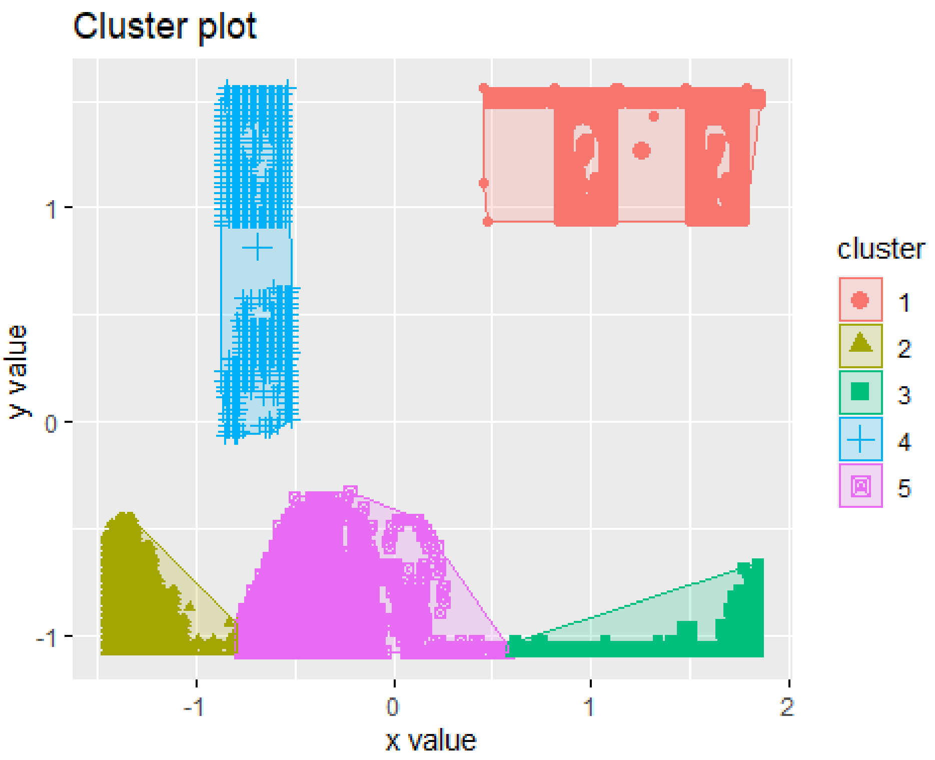
```
kmean3 <- kmeans(cluster_df, 3,)
fviz_cluster(kmean3, geom = "point", data = cluster_df)
```



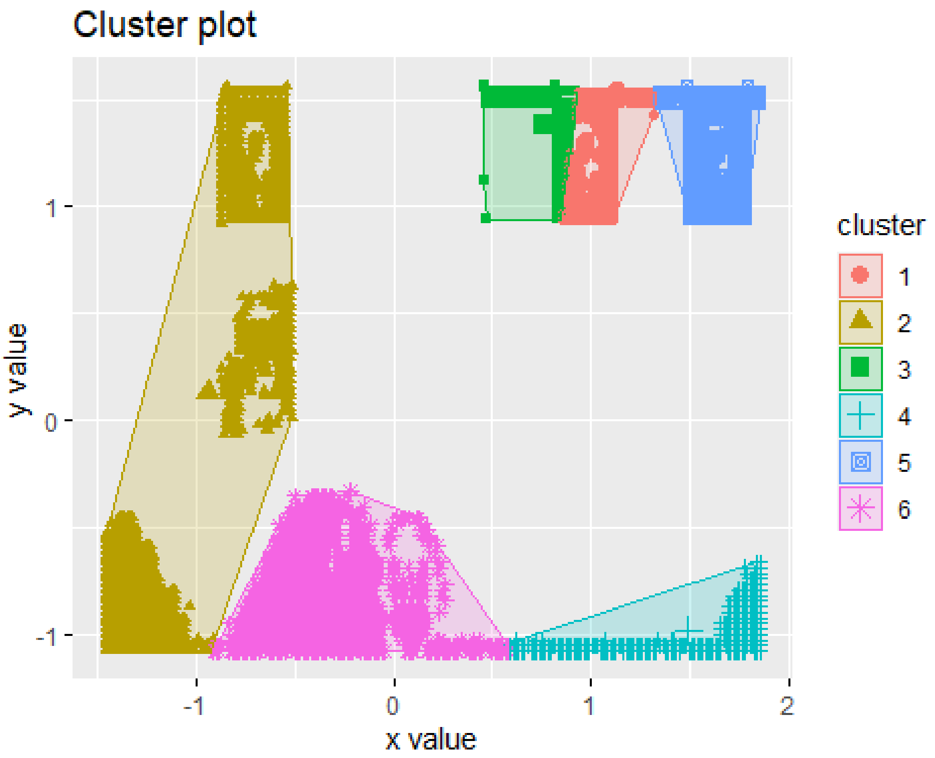
```
kmean4 <- kmeans(cluster_df, 4,)
fviz_cluster(kmean4, geom = "point", data = cluster_df)
```



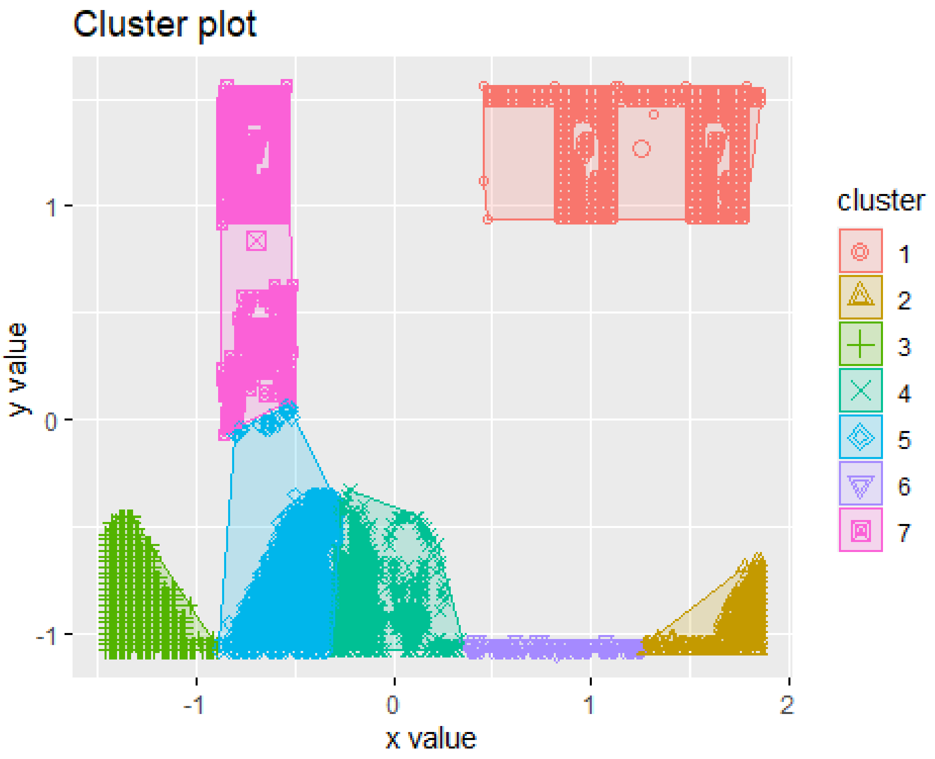
```
kmean5 <- kmeans(cluster_df, 5,)
fviz_cluster(kmean5, geom = "point", data = cluster_df)
```



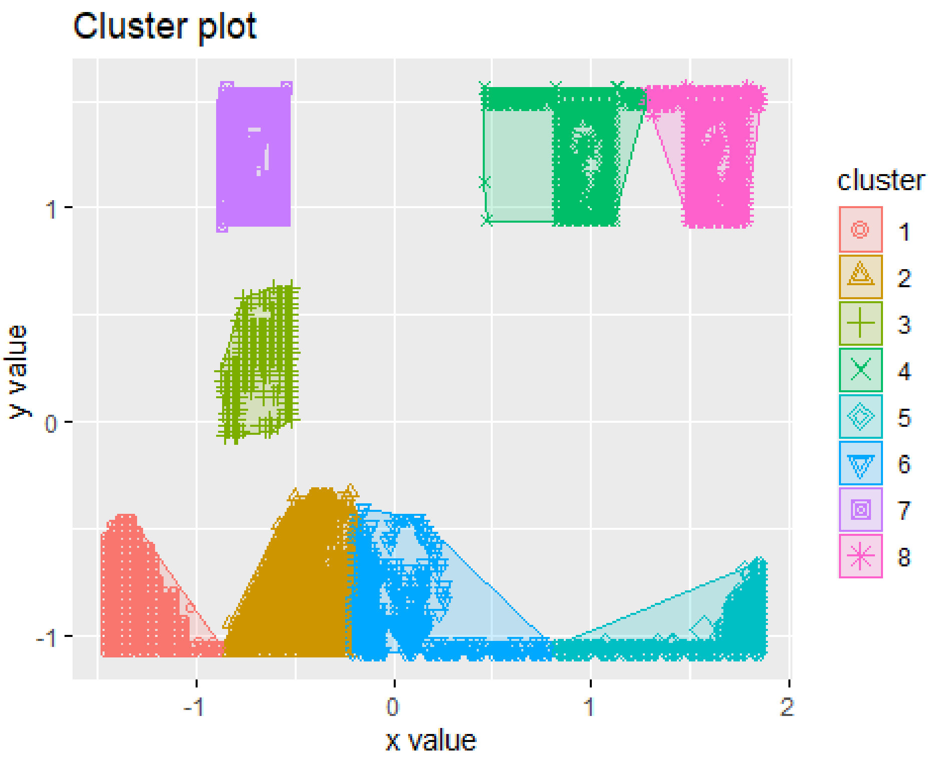
```
kmean6 <- kmeans(cluster_df, 6,)
fviz_cluster(kmean6, geom = "point", data = cluster_df)
```



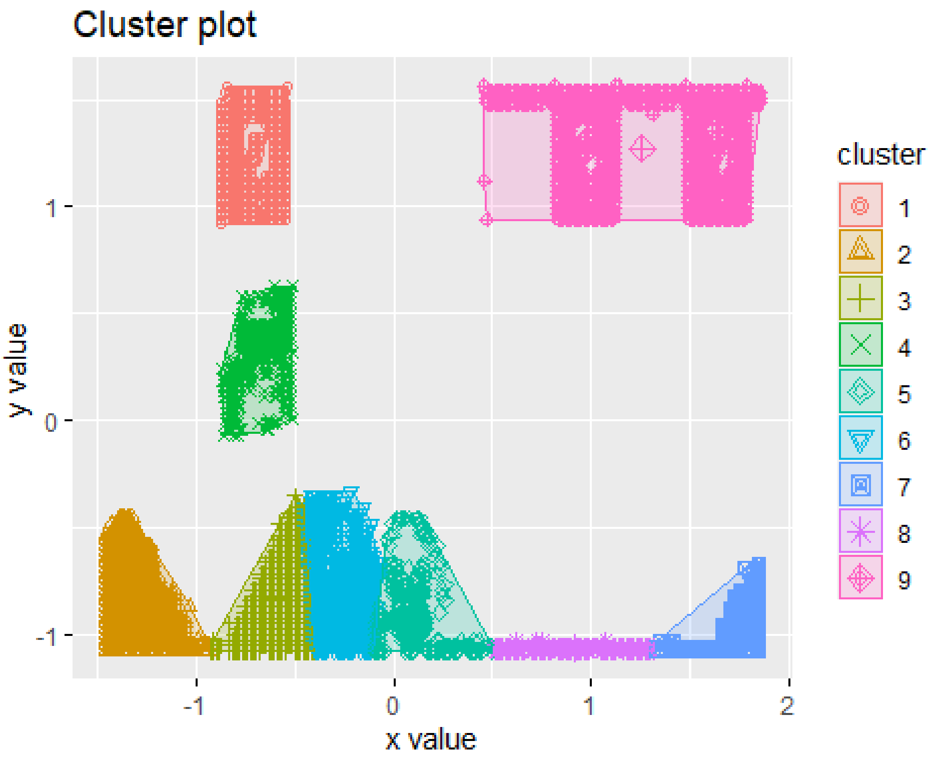
```
kmean7 <- kmeans(cluster_df, 7,)
fviz_cluster(kmean7, geom = "point", data = cluster_df)
```



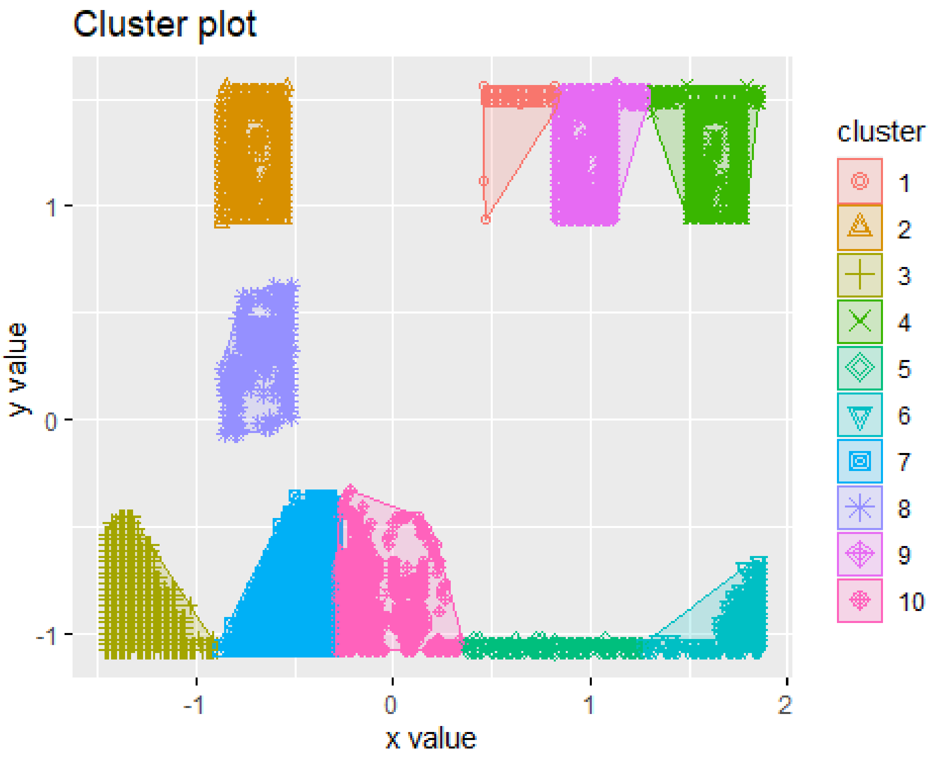
```
kmean8 <- kmeans(cluster_df, 8,)
fviz_cluster(kmean8, geom = "point", data = cluster_df)
```



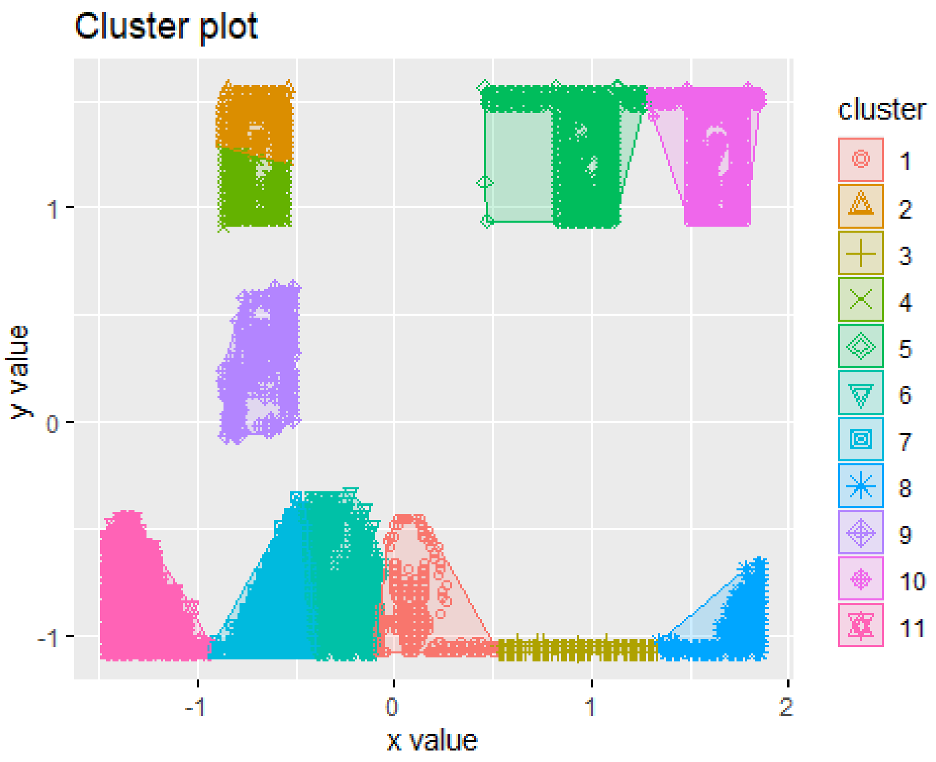
```
kmean9 <- kmeans(cluster_df, 9,)
fviz_cluster(kmean9, geom = "point", data = cluster_df)
```



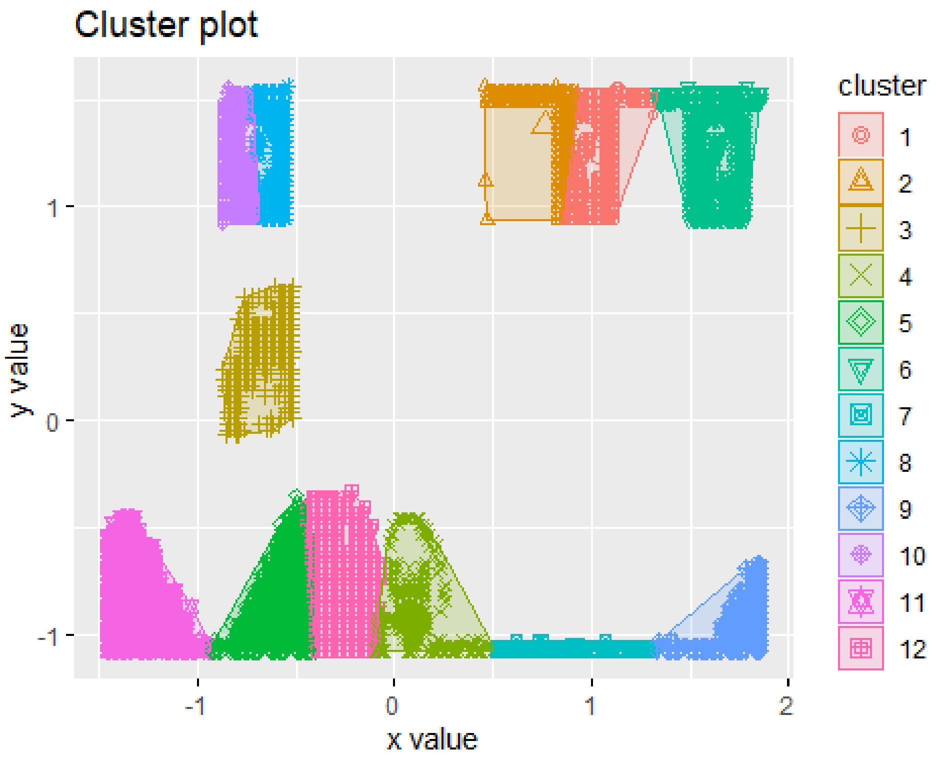
```
kmean10 <- kmeans(cluster_df, 10,)
fviz_cluster(kmean10, geom = "point", data = cluster_df)
```



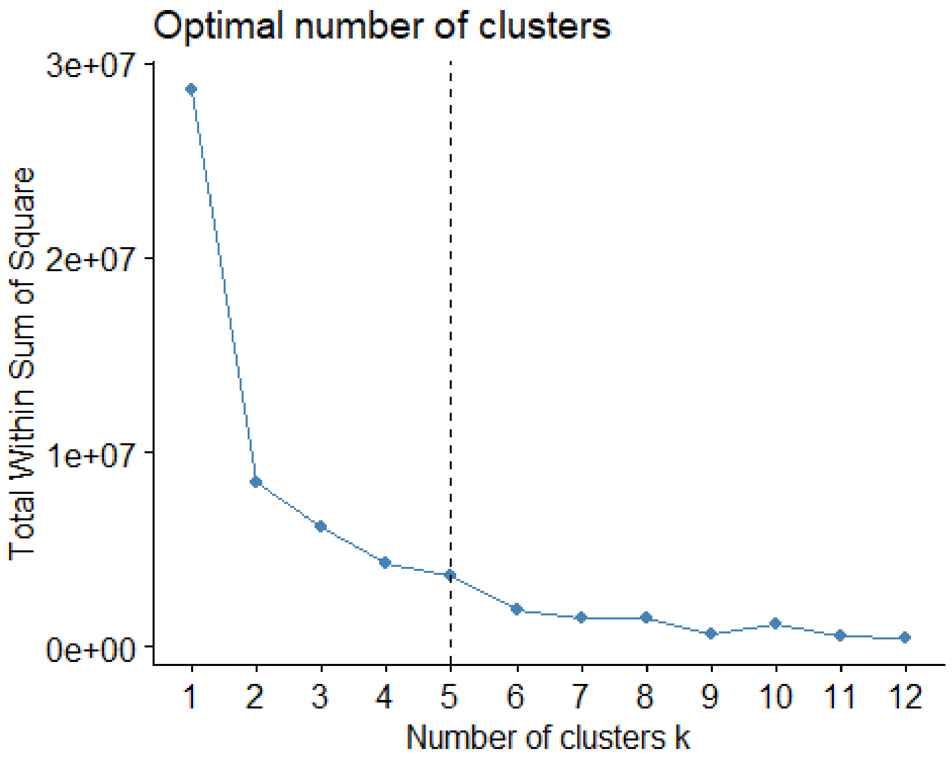
```
kmean11 <- kmeans(cluster_df, 11,)
fviz_cluster(kmean11, geom = "point", data = cluster_df)
```



```
kmean12 <- kmeans(cluster_df, 12,)
fviz_cluster(kmean12, geom = "point", data = cluster_df)
```



```
#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.


fviz_nbclust(cluster_df, kmeans, method = "wss", k.max = 12) +
  geom_vline(xintercept = 5, linetype = 2)
```



The elbow point seems to be at 4 or 5 clusters