ASSIGNMENT 11.2.2

2022-06-04

Installing Packages

install.packages("e1071") install.packages("caTools") install.packages("class") install.packages("tidymodels") install.packages("gridExtra") install.packages("kknn") install.packages("factoextra") install.packages("cluster")

```
# Loading package
library(e1071)
library(caTools)
library(class)
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 0.2.0 --
```

```
## v broom        0.7.12     v rsample      0.1.1
## v dials        0.1.1      v tune         0.2.0
## v infer        1.0.0      v workflows    0.2.6
## v modeldata    0.1.1      v workflowsets 0.2.1
## v parsnip      0.2.1      v yardstick    0.0.9
## v recipes      0.2.0
```

```
## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard()       masks purrr::discard()
## x dplyr::filter()         masks stats::filter()
## x recipes::fixed()        masks stringr::fixed()
## x dplyr::lag()            masks stats::lag()
## x rsample::permutations() masks e1071::permutations()
## x yardstick::spec()       masks readr::spec()
## x recipes::step()         masks stats::step()
## x tune::tune()            masks parsnip::tune(), e1071::tune()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(ggplot2)
library(kknn)
library(factoextra)  # clustering algorithms & visualization
```

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

```
library(cluster)  # clustering algorithms

set.seed(12)


## Load the `data/clustering-data.csv` to
clstr_df <- read.csv("/Users/siddharthabhaumik/Documents/GitHub/dsc520/data/clustering-data.csv")

## Viewing Sample data
head(clstr_df)
```

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

```
## summary
summary(clstr_df)
```

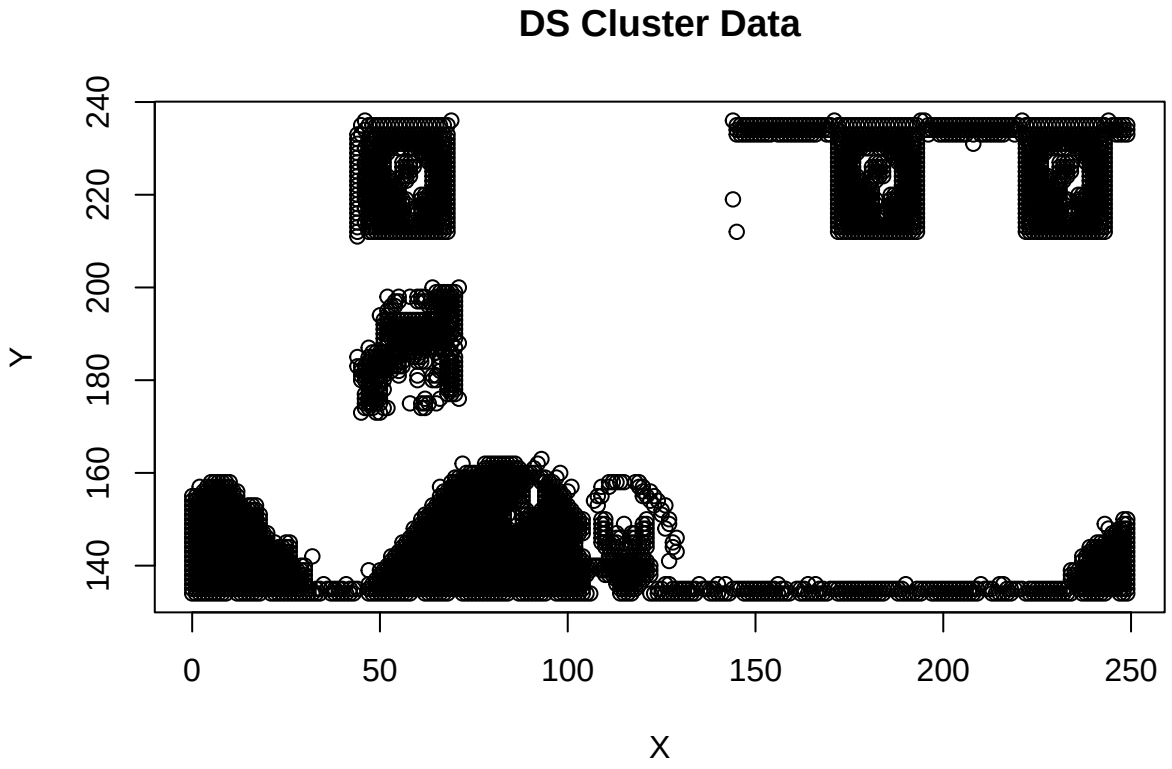
```
##        x               y
##  Min.   :  0.0   Min.   :134.0
##  1st Qu.: 56.0   1st Qu.:141.0
##  Median : 82.0   Median :154.0
##  Mean   :109.6   Mean   :175.7
```

```
##  3rd Qu.:180.0   3rd Qu.:218.0
##  Max.   :249.0   Max.   :236.0
```

```
## Check Data Structure of the object
str(clstr_df)
```

```
## 'data.frame':    4022 obs. of  2 variables:
##  $ x: int  46 69 144 171 194 195 221 244 45 47 ...
##  $ y: int  236 236 236 236 236 236 236 236 235 235 ...
```

```
# Plot the data from dataset using a scatter plot.

plot(clstr_df$x,clstr_df$y,main = "DS Cluster Data", xlab = "X", ylab = "Y")
```

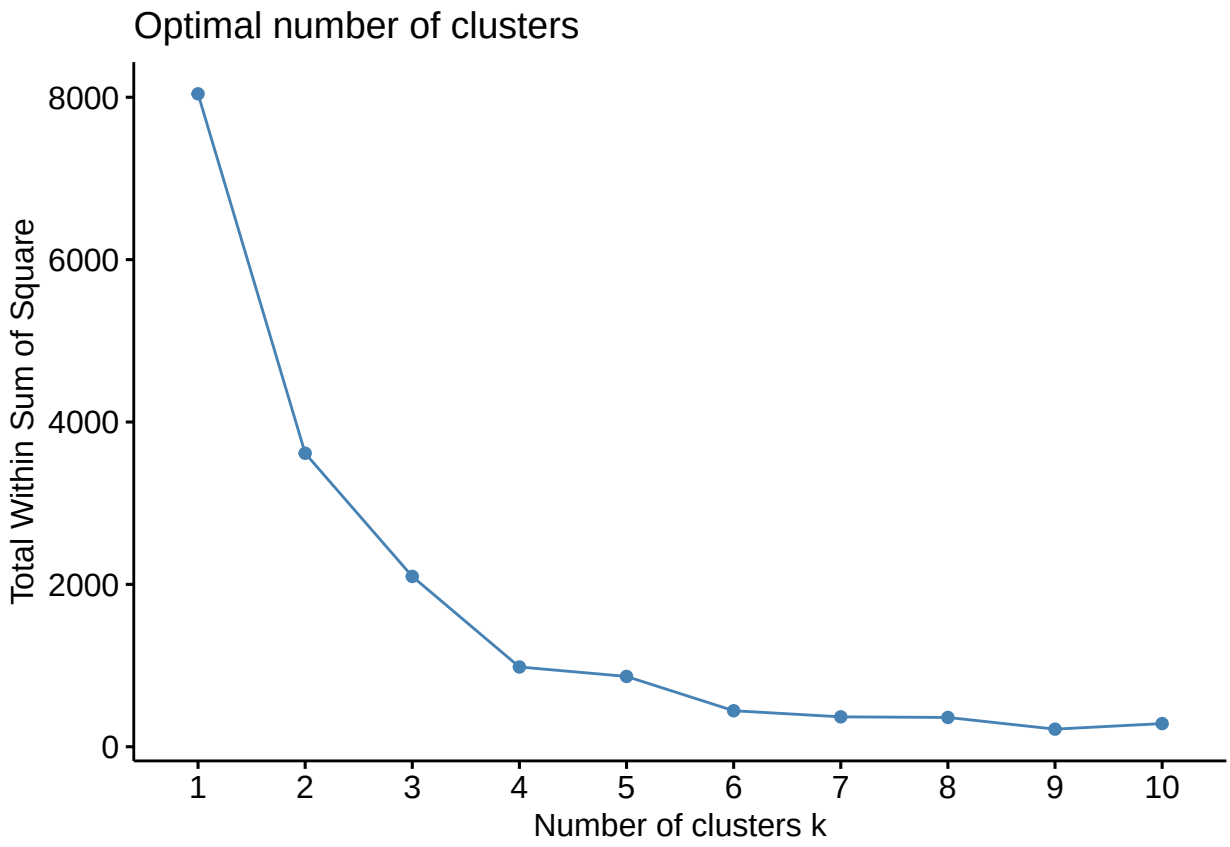



```
#remove rows with missing values
clstr_upd_df <- na.omit(clstr_df)

#scale each variable to have a mean of 0 and sd of 1
clstr_upd_df <- scale(clstr_upd_df)

head(clstr_upd_df)
```

```
##            x        y
## 1 -0.8482235 1.561107
## 2 -0.5415045 1.561107
## 3  0.4586659 1.561107
## 4  0.8187273 1.561107
## 5  1.1254462 1.561107
## 6  1.1387818 1.561107
```

```
#DETERMINE HOW MANY CLUSTERS IS OPTIMAL
#plot number of clusters vs. total within sum of squares
```

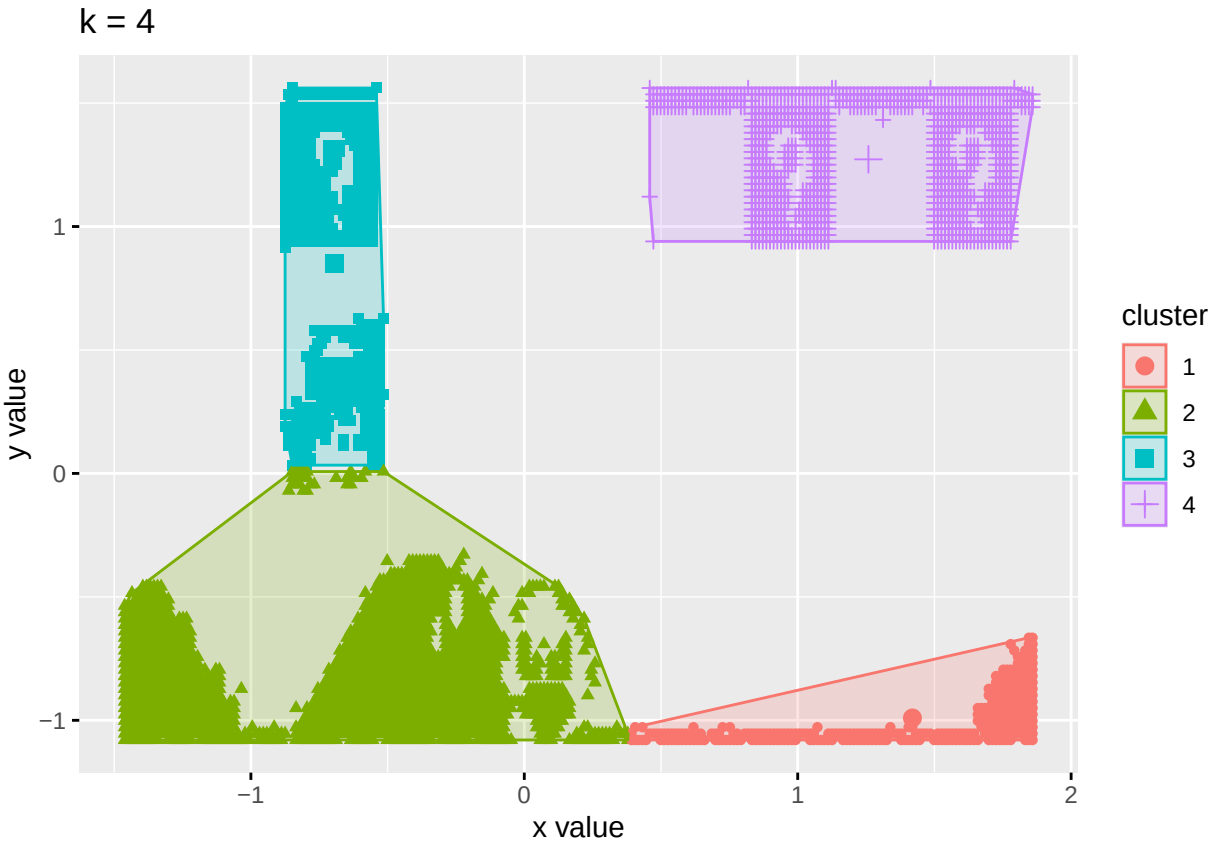
```
fviz_nbclust(clstr_upd_df, kmeans, method = "wss")
```




```
# For this plot it appear that there is a bit of an elbow or "bend" at k = 4 clusters.

# Perform K-Means Clustering with Optimal K(i.e.with k = 4 clusters)
# nstart = 25 will generate 25 initial configurations. This approach is often recommended.

clstr_km <- kmeans(clstr_upd_df, centers = 4, nstart = 25)

#view results
#head(clstr_km)

# Cluster center
clstr_km$centers
```

```
##           x          y
## 1  1.4196187 -0.9906559
## 2 -0.6150938 -0.8013866
## 3 -0.6939089  0.8493881
## 4  1.2586128  1.2721827
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 4")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
head(final_data)
```

```
##            x        y cluster
## 1 -0.8482235 1.561107       3
## 2 -0.5415045 1.561107       3
## 3  0.4586659 1.561107       4
## 4  0.8187273 1.561107       4
## 5  1.1254462 1.561107       4
## 6  1.1387818 1.561107       4
```

```
#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

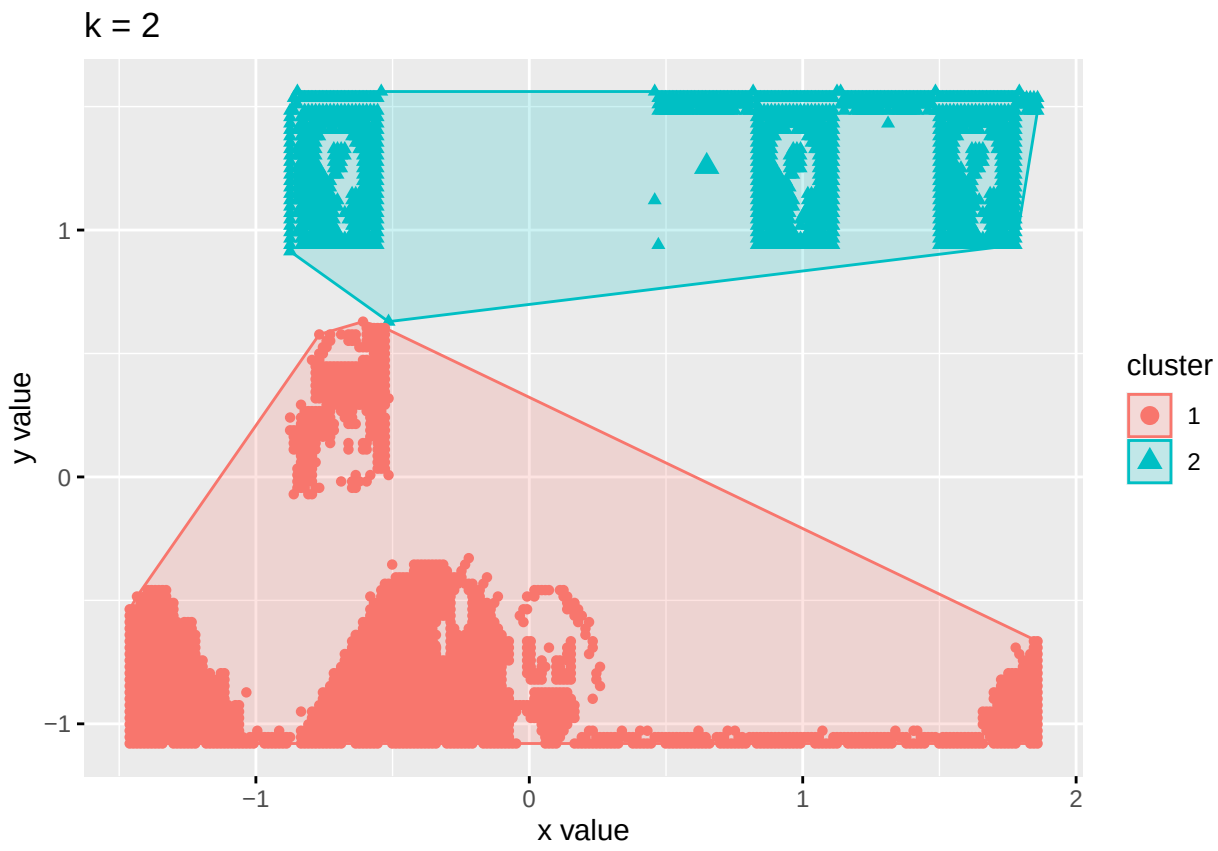
```
##   cluster          x          y
## 1       1  1.4196187 -0.9906559
## 2       2 -0.6150938 -0.8013866
## 3       3 -0.6939089  0.8493881
## 4       4  1.2586128  1.2721827
```

```
# Perform K-Means Clustering with other K value 2

clstr_km <- kmeans(clstr_upd_df, centers = 2, nstart = 25)

#view results
#head(clstr_km)
```

```
# Cluster center
clstr_km$centers
```

```
##           x          y
## 1 -0.3568359 -0.6913114
## 2  0.6489063  1.2571500
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 2")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
head(final_data)
```

```
##           x        y cluster
## 1 -0.8482235 1.561107       2
## 2 -0.5415045 1.561107       2
## 3  0.4586659 1.561107       2
## 4  0.8187273 1.561107       2
## 5  1.1254462 1.561107       2
## 6  1.1387818 1.561107       2
```

```
#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

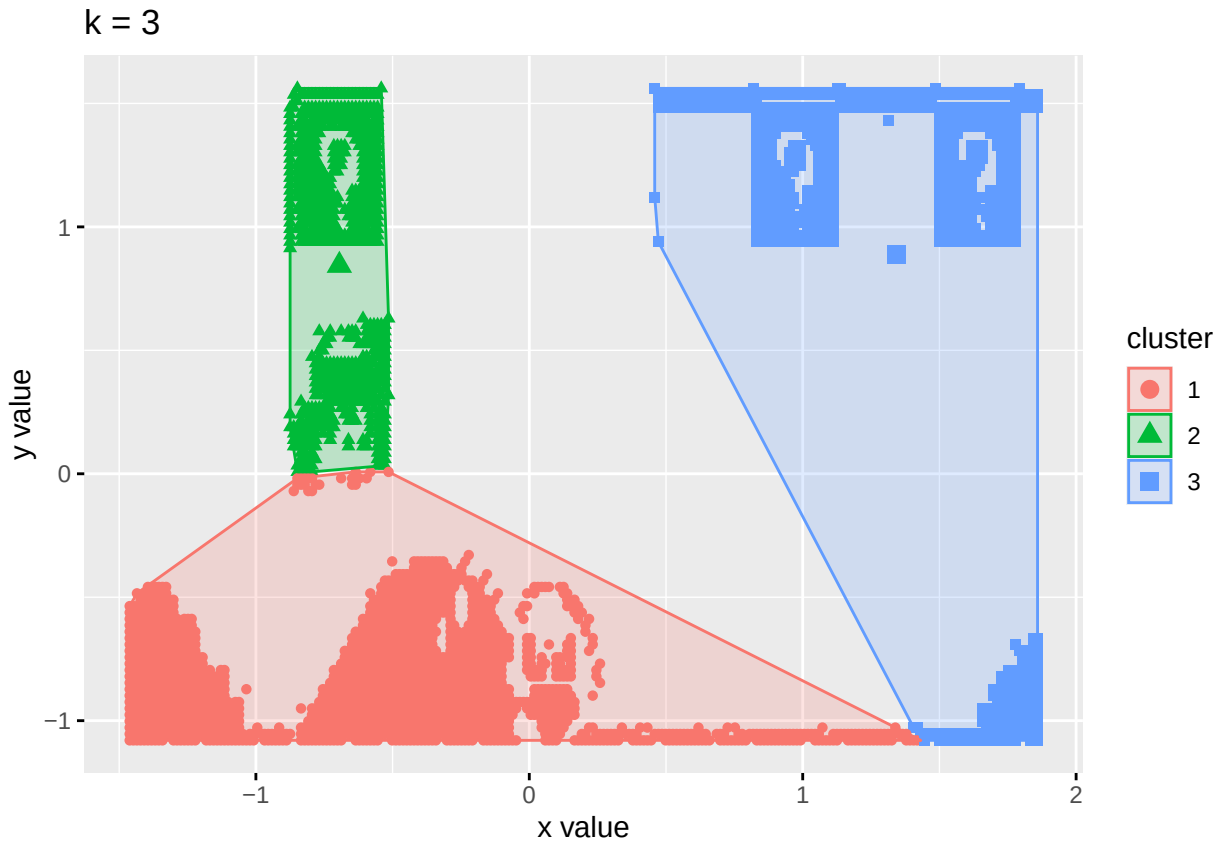
```
##   cluster          x          y
```

```
## 1       1 -0.3568359 -0.6913114
## 2       2  0.6489063  1.2571500
```

```
# Perform K-Means Clustering with other K value 3

clstr_km <- kmeans(clstr_upd_df, centers = 3, nstart = 25)

#view results
#head(clstr_km)

# Cluster center
clstr_km$centers
```

```
##           x          y
## 1 -0.5183491 -0.8200220
## 2 -0.6947498  0.8438432
## 3  1.3429665  0.8874064
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 3")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
head(final_data)
```

```
##           x        y cluster
## 1 -0.8482235 1.561107       2
```

```
## 2 -0.5415045 1.561107       2
## 3  0.4586659 1.561107       3
## 4  0.8187273 1.561107       3
## 5  1.1254462 1.561107       3
## 6  1.1387818 1.561107       3
```

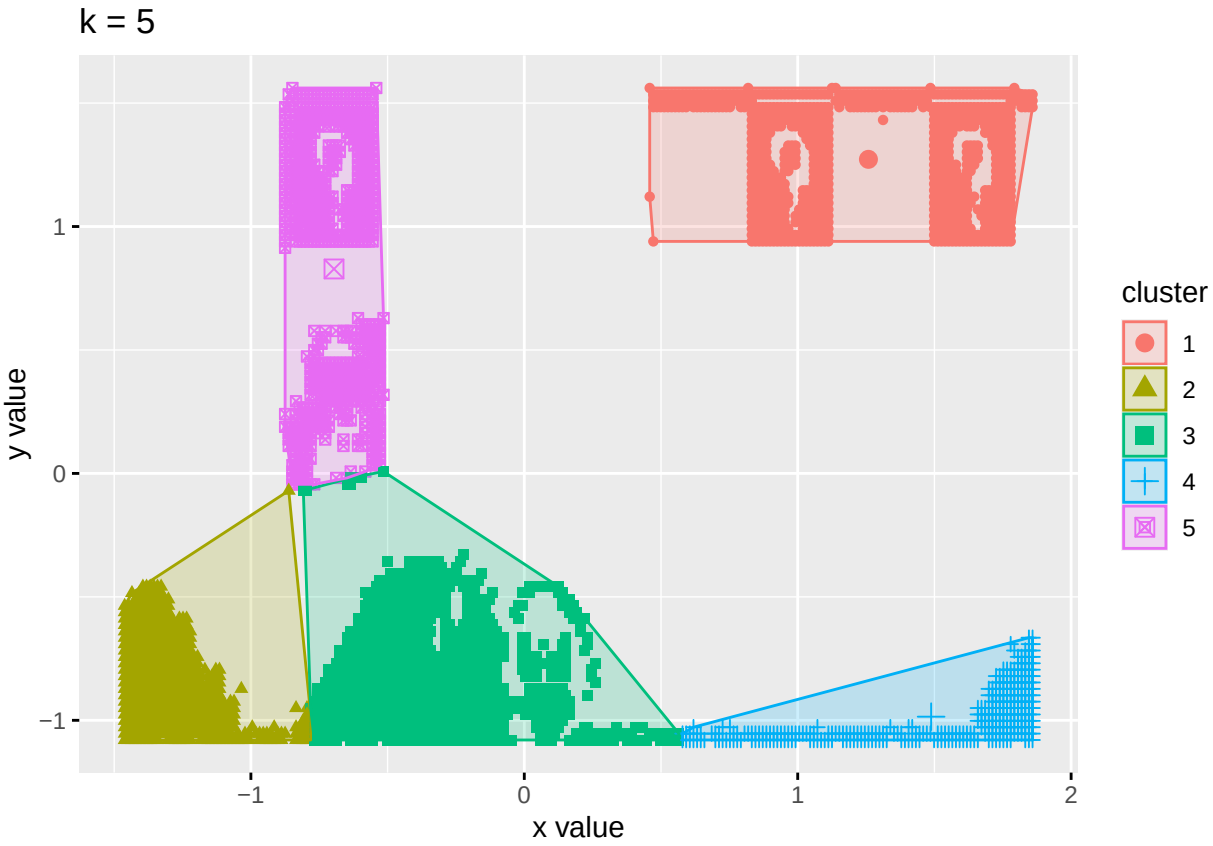
```
#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

```
##   cluster          x          y
## 1       1 -0.5183491 -0.8200220
## 2       2 -0.6947498  0.8438432
## 3       3  1.3429665  0.8874064
```

```
# Perform K-Means Clustering with other K value 5

clstr_km <- kmeans(clstr_upd_df, centers = 5, nstart = 25)

#view results
#head(clstr_km)

# Cluster center
clstr_km$centers
```

```
##           x          y
## 1  1.2586128  1.2721827
## 2 -1.2687005 -0.8461129
## 3 -0.3090208 -0.7973288
## 4  1.4879630 -0.9853019
## 5 -0.6960285  0.8281296
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 5")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
head(final_data)
```

```
##           x        y cluster
## 1 -0.8482235 1.561107       5
## 2 -0.5415045 1.561107       5
## 3  0.4586659 1.561107       1
## 4  0.8187273 1.561107       1
## 5  1.1254462 1.561107       1
## 6  1.1387818 1.561107       1
```

```
#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

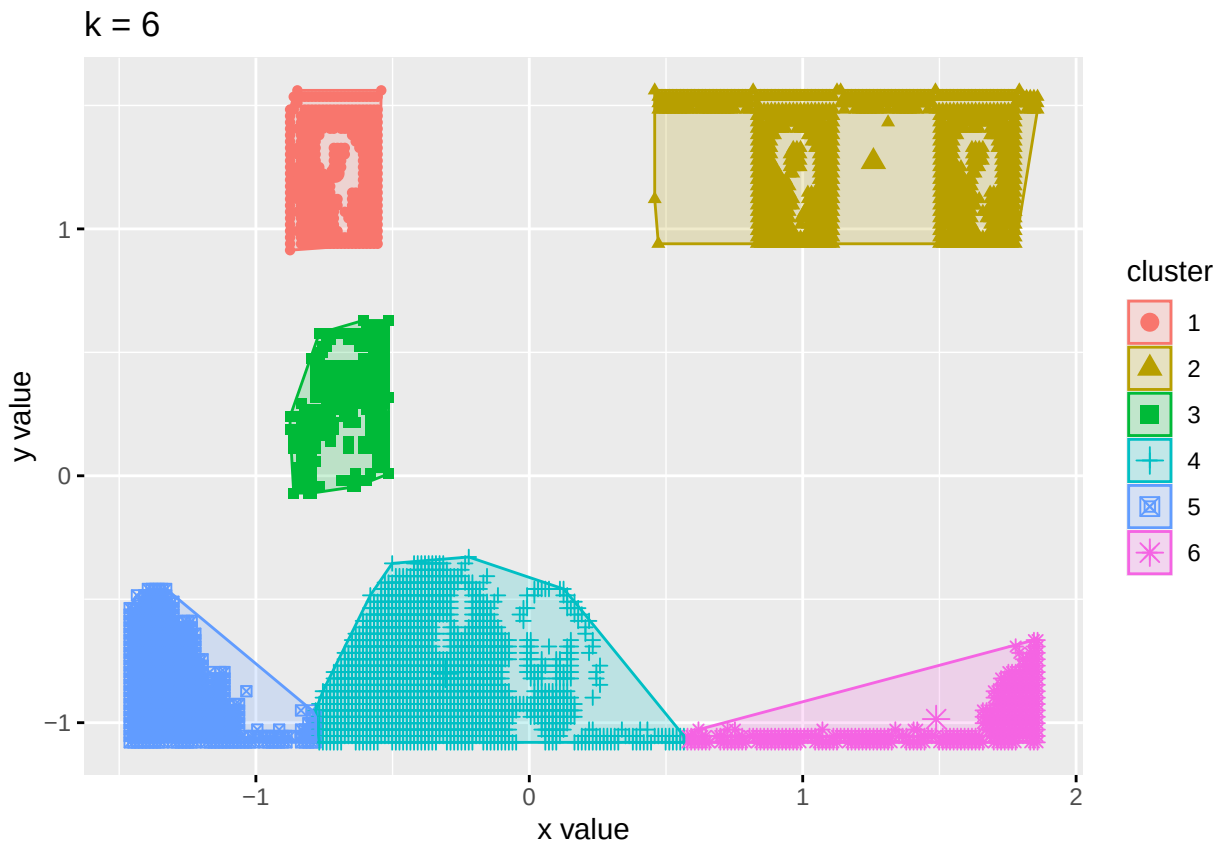
```
##   cluster          x          y
## 1       1  1.2586128  1.2721827
## 2       2 -1.2687005 -0.8461129
## 3       3 -0.3090208 -0.7973288
## 4       4  1.4879630 -0.9853019
## 5       5 -0.6960285  0.8281296
```

```
# Perform K-Means Clustering with other K value 6

clstr_km <- kmeans(clstr_upd_df, centers = 6, nstart = 25)

#view results
```

```
#head(clstr_km)

# Cluster center
clstr_km$centers
```

```
##           x          y
## 1 -0.7102710  1.2249978
## 2  1.2586128  1.2721827
## 3 -0.6771679  0.2920086
## 4 -0.3069600 -0.8018532
## 5 -1.2693894 -0.8474261
## 6  1.4879630 -0.9853019
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 6")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
head(final_data)
```

```
##           x        y cluster
## 1 -0.8482235 1.561107       1
## 2 -0.5415045 1.561107       1
## 3  0.4586659 1.561107       2
## 4  0.8187273 1.561107       2
## 5  1.1254462 1.561107       2
```

```
## 6  1.1387818 1.561107       2
```

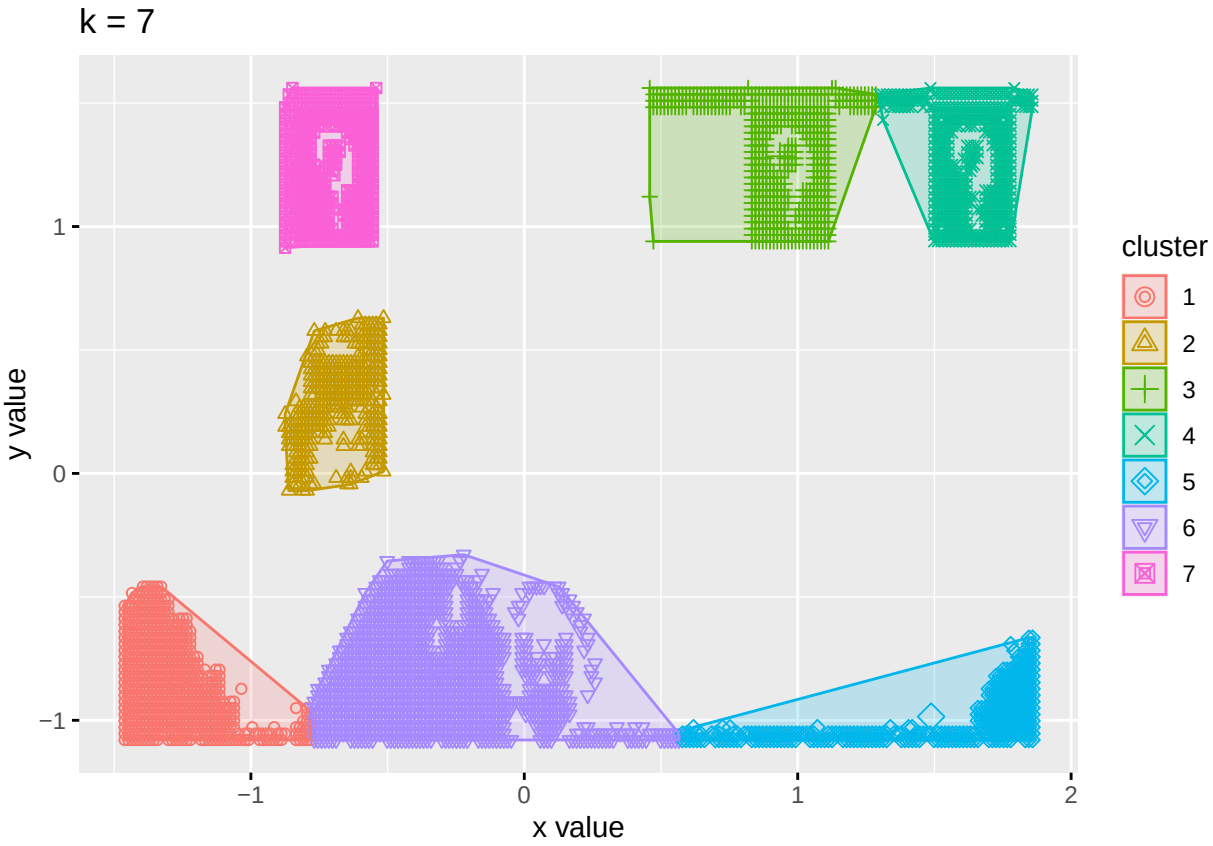
```
#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

```
##   cluster          x          y
## 1       1 -0.7102710  1.2249978
## 2       2  1.2586128  1.2721827
## 3       3 -0.6771679  0.2920086
## 4       4 -0.3069600 -0.8018532
## 5       5 -1.2693894 -0.8474261
## 6       6  1.4879630 -0.9853019
```

```
# Perform K-Means Clustering with other K value 7

clstr_km <- kmeans(clstr_upd_df, centers = 7, nstart = 25)

#view results
#head(clstr_km)

# Cluster center
clstr_km$centers
```

```
##           x          y
## 1 -1.2693894 -0.8474261
## 2 -0.6771679  0.2920086
## 3  0.9347803  1.2845090
## 4  1.6222263  1.2583422
## 5  1.4879630 -0.9853019
## 6 -0.3069600 -0.8018532
## 7 -0.7102710  1.2249978
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 7")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
head(final_data)
```

```
##           x        y cluster
## 1 -0.8482235 1.561107       7
## 2 -0.5415045 1.561107       7
## 3  0.4586659 1.561107       3
## 4  0.8187273 1.561107       3
## 5  1.1254462 1.561107       3
## 6  1.1387818 1.561107       3
```

```
#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

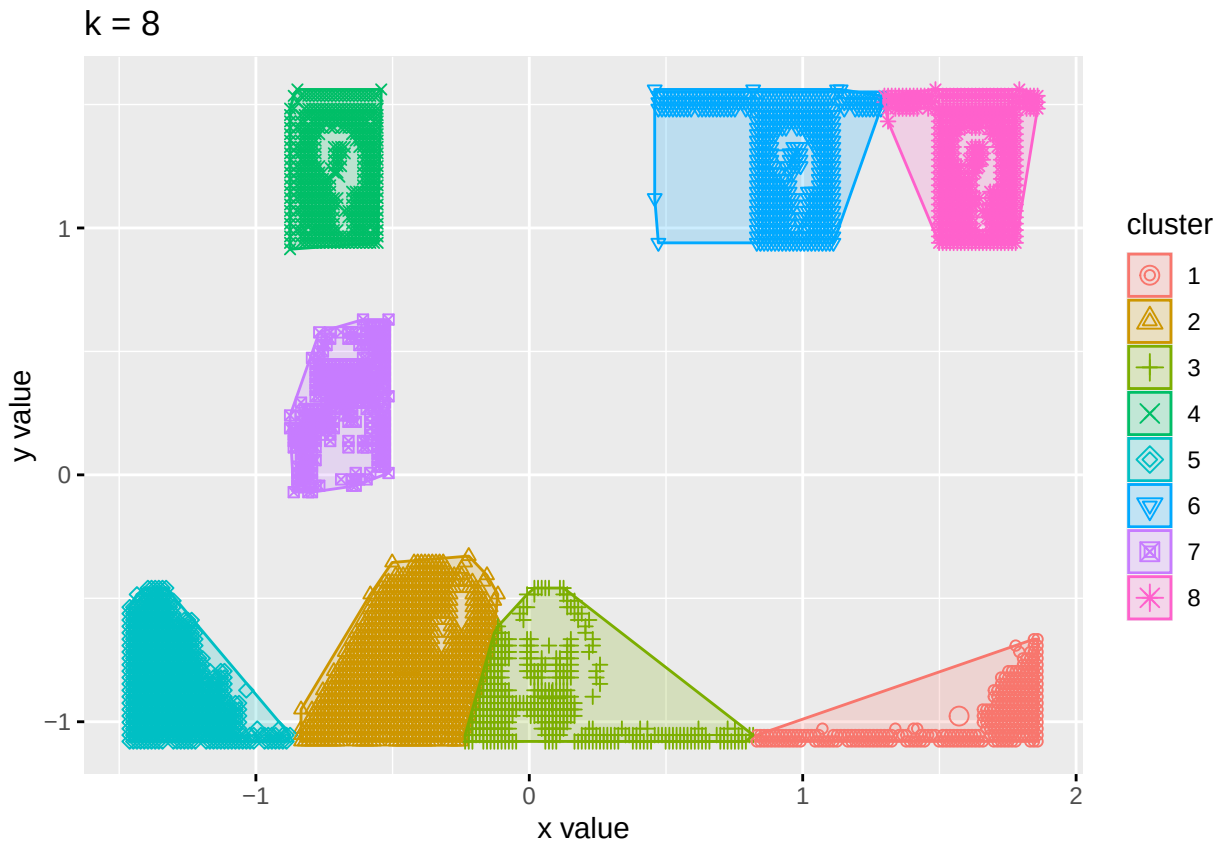
```
##   cluster          x          y
## 1       1 -1.2693894 -0.8474261
## 2       2 -0.6771679  0.2920086
## 3       3  0.9347803  1.2845090
## 4       4  1.6222263  1.2583422
## 5       5  1.4879630 -0.9853019
## 6       6 -0.3069600 -0.8018532
## 7       7 -0.7102710  1.2249978
```

```
# Perform K-Means Clustering with other K value 8

clstr_km <- kmeans(clstr_upd_df, centers = 8, nstart = 25)
```

```
#view results
#head(clstr_km)

# Cluster center
clstr_km$centers
```

```
##             x          y
## 1  1.57193168 -0.9769962
## 2 -0.44740397 -0.7724133
## 3  0.08261495 -0.9054404
## 4 -0.71027100  1.2249978
## 5 -1.28733754 -0.8403549
## 6  0.93478034  1.2845090
## 7 -0.67716791  0.2920086
## 8  1.62222630  1.2583422
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 8")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
head(final_data)
```

```
##            x        y cluster
## 1 -0.8482235 1.561107       4
```

```
## 2 -0.5415045 1.561107       4
## 3  0.4586659 1.561107       6
## 4  0.8187273 1.561107       6
## 5  1.1254462 1.561107       6
## 6  1.1387818 1.561107       6
```

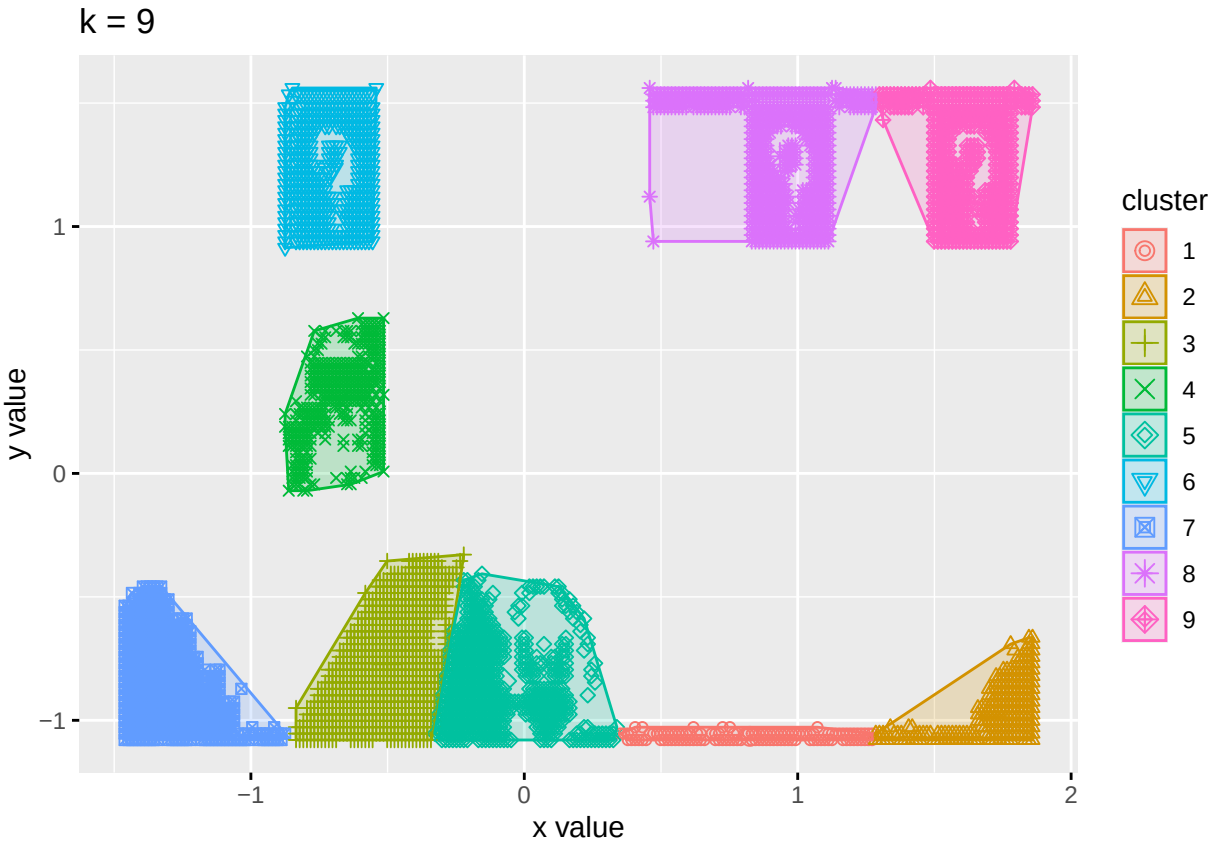
```
#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

```
##   cluster           x          y
## 1       1  1.57193168 -0.9769962
## 2       2 -0.44740397 -0.7724133
## 3       3  0.08261495 -0.9054404
## 4       4 -0.71027100  1.2249978
## 5       5 -1.28733754 -0.8403549
## 6       6  0.93478034  1.2845090
## 7       7 -0.67716791  0.2920086
## 8       8  1.62222630  1.2583422
```

```
# Perform K-Means Clustering with other K value 9

clstr_km <- kmeans(clstr_upd_df, centers = 9, nstart = 25)

#view results
#head(clstr_km)

# Cluster center
clstr_km$centers
```

```
##             x          y
## 1  0.82471702 -1.0648212
## 2  1.71423705 -0.9528307
## 3 -0.50046805 -0.7729546
## 4 -0.67716791  0.2920086
## 5 -0.07860369 -0.8409824
## 6 -0.71027100  1.2249978
## 7 -1.28733754 -0.8403549
## 8  0.93478034  1.2845090
## 9  1.62222630  1.2583422
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 9")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
#head(final_data)

#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

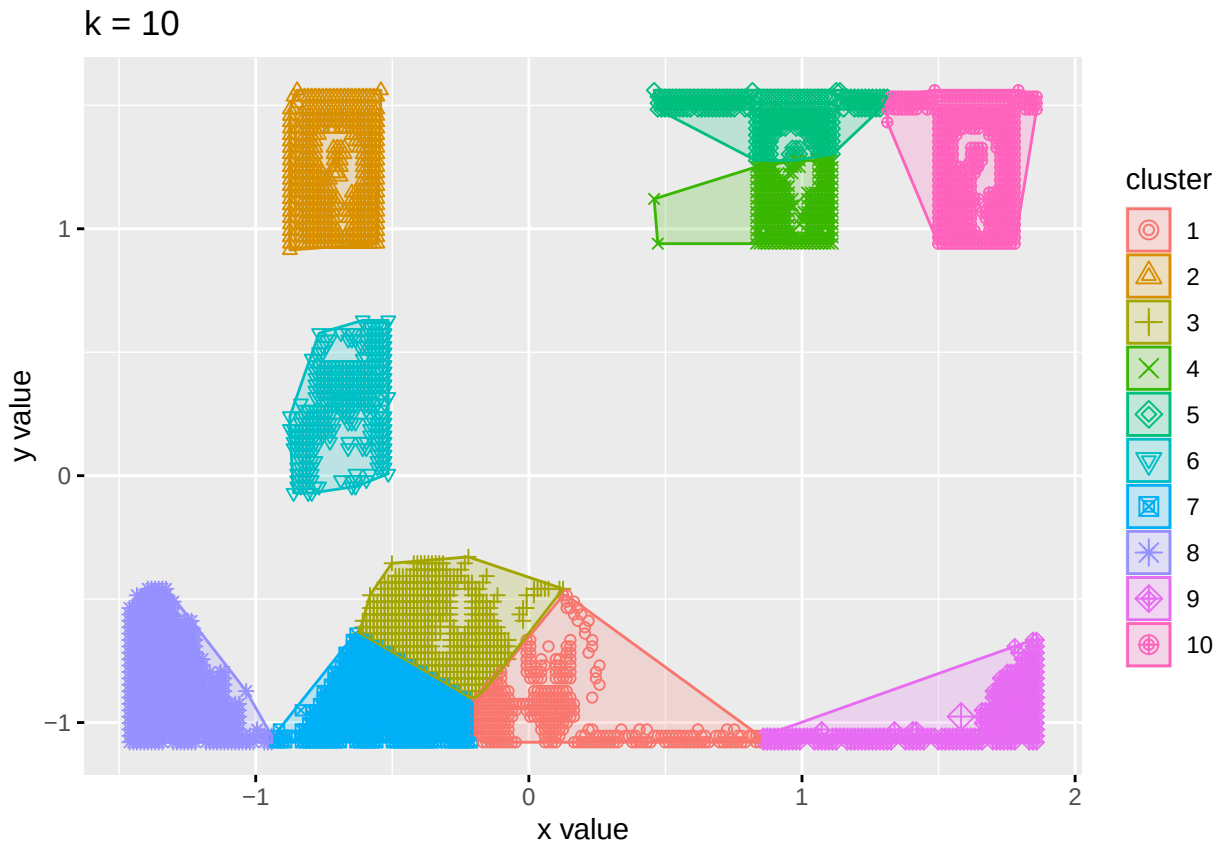
```
##   cluster           x          y
## 1       1  0.82471702 -1.0648212
## 2       2  1.71423705 -0.9528307
## 3       3 -0.50046805 -0.7729546
## 4       4 -0.67716791  0.2920086
## 5       5 -0.07860369 -0.8409824
## 6       6 -0.71027100  1.2249978
## 7       7 -1.28733754 -0.8403549
## 8       8  0.93478034  1.2845090
## 9       9  1.62222630  1.2583422
```

```
# Perform K-Means Clustering with other K value 10

clstr_km <- kmeans(clstr_upd_df, centers = 10, nstart = 25)

#view results
#head(clstr_km)

# Cluster center
```

```
clstr_km$centers
```

```
##             x          y
## 1   0.1338769 -0.9414023
## 2  -0.7102710  1.2249978
## 3  -0.3244840 -0.6111989
## 4   0.9663558  1.0880137
## 5   0.9116036  1.4557079
## 6  -0.6771679  0.2920086
## 7  -0.5280962 -0.9280690
## 8  -1.2934898 -0.8367851
## 9   1.5823687 -0.9757179
## 10  1.6243020  1.2565966
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 10")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
#head(final_data)

#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

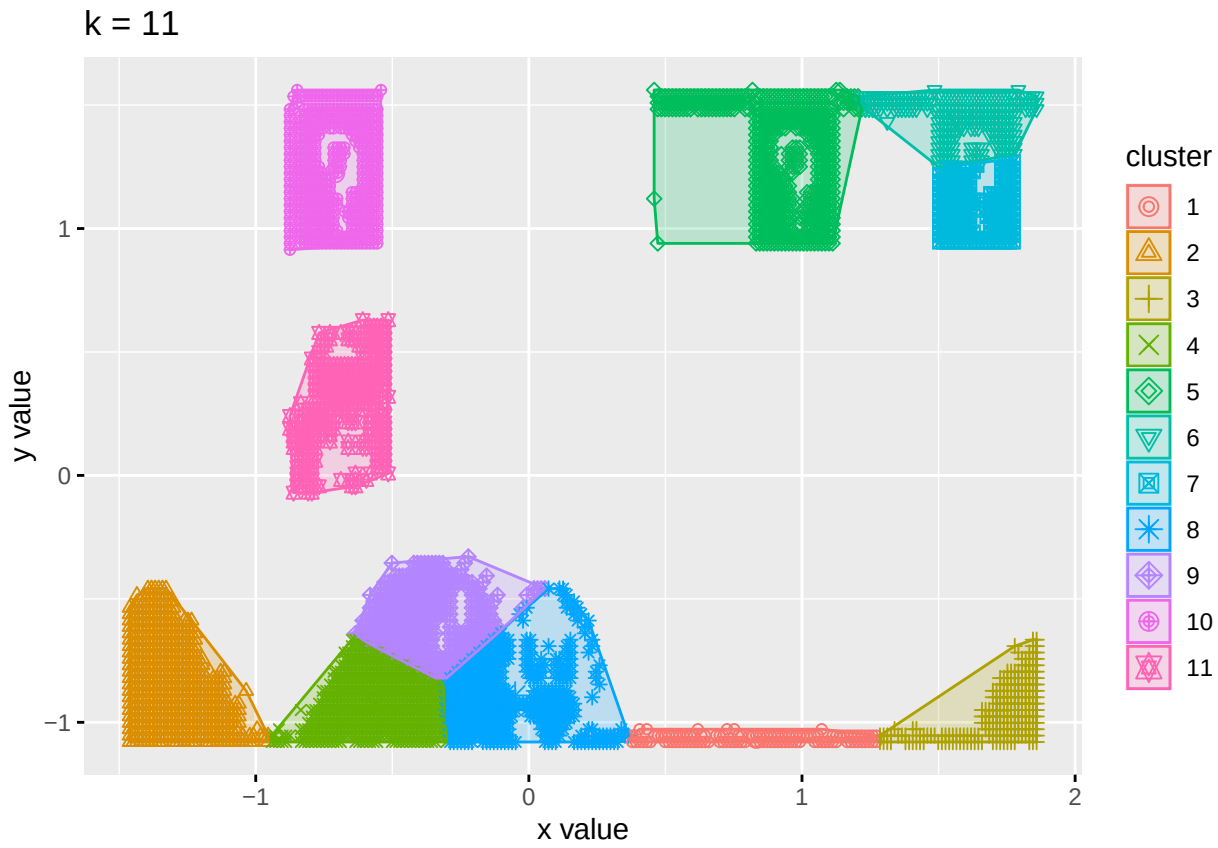
```
##    cluster          x          y
## 1        1  0.1338769 -0.9414023
```

```
## 2        2 -0.7102710  1.2249978
## 3        3 -0.3244840 -0.6111989
## 4        4  0.9663558  1.0880137
## 5        5  0.9116036  1.4557079
## 6        6 -0.6771679  0.2920086
## 7        7 -0.5280962 -0.9280690
## 8        8 -1.2934898 -0.8367851
## 9        9  1.5823687 -0.9757179
## 10      10  1.6243020  1.2565966
```

```
# Perform K-Means Clustering with other K value 11

clstr_km <- kmeans(clstr_upd_df, centers = 11, nstart = 25)

#view results
#head(clstr_km)

# Cluster center
clstr_km$centers
```

```
##              x          y
## 1   0.83275268 -1.0650103
## 2  -1.29475117 -0.8359606
## 3   1.71423705 -0.9528307
## 4  -0.56373971 -0.9251996
## 5   0.92400979  1.2768745
## 6   1.58142558  1.4427949
## 7   1.63782610  1.0865538
## 8  -0.05021836 -0.8912359
## 9  -0.36519861 -0.5777004
## 10 -0.71027100  1.2249978
## 11 -0.67716791  0.2920086
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 11")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
#head(final_data)

#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

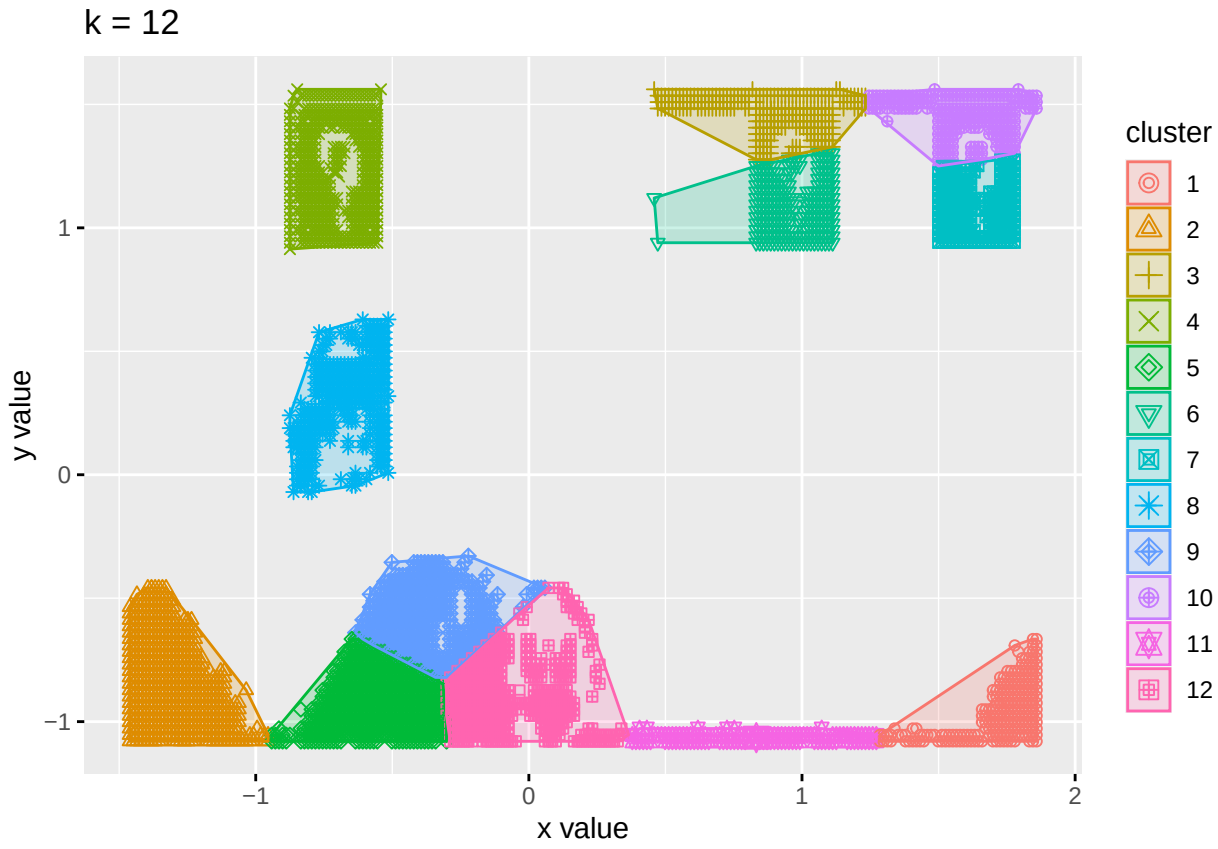
```
##    cluster           x          y
## 1        1  0.83275268 -1.0650103
## 2        2 -1.29475117 -0.8359606
## 3        3  1.71423705 -0.9528307
## 4        4 -0.56373971 -0.9251996
## 5        5  0.92400979  1.2768745
## 6        6  1.58142558  1.4427949
## 7        7  1.63782610  1.0865538
## 8        8 -0.05021836 -0.8912359
## 9        9 -0.36519861 -0.5777004
## 10      10 -0.71027100  1.2249978
## 11      11 -0.67716791  0.2920086
```

```
# Perform K-Means Clustering with other K value 12

clstr_km <- kmeans(clstr_upd_df, centers = 12, nstart = 25)

#view results
#head(clstr_km)
```

```
# Cluster center
clstr_km$centers
```

```
##              x          y
## 1   1.71423705 -0.9528307
## 2  -1.29475117 -0.8359606
## 3   0.88704663  1.4567475
## 4  -0.71027100  1.2249978
## 5  -0.56373971 -0.9251996
## 6   0.96868592  1.0938270
## 7   1.63729817  1.0872423
## 8  -0.67716791  0.2920086
## 9  -0.36519861 -0.5777004
## 10  1.58916672  1.4420967
## 11  0.83275268 -1.0650103
## 12 -0.05021836 -0.8912359
```

```
#plot results of final k-means model
fviz_cluster(clstr_km, geom = "point",data = clstr_upd_df) + ggtitle("k = 12")
```




```
#add cluster assignment to original data
final_data <- cbind(clstr_upd_df, cluster = clstr_km$cluster)

#view final data
head(final_data)
```

```
##            x        y cluster
```

```
## 1 -0.8482235 1.561107       4
## 2 -0.5415045 1.561107       4
## 3  0.4586659 1.561107       3
## 4  0.8187273 1.561107       3
## 5  1.1254462 1.561107       3
## 6  1.1387818 1.561107       3
```

```
#find means of each cluster
aggregate(clstr_upd_df, by=list(cluster=clstr_km$cluster), mean)
```

```
##    cluster           x          y
## 1        1  1.71423705 -0.9528307
## 2        2 -1.29475117 -0.8359606
## 3        3  0.88704663  1.4567475
## 4        4 -0.71027100  1.2249978
## 5        5 -0.56373971 -0.9251996
## 6        6  0.96868592  1.0938270
## 7        7  1.63729817  1.0872423
## 8        8 -0.67716791  0.2920086
## 9        9 -0.36519861 -0.5777004
## 10      10  1.58916672  1.4420967
## 11      11  0.83275268 -1.0650103
## 12      12 -0.05021836 -0.8912359
```