RMarkdown Assignment # 16 - Week 09

Pushkar Chougule

Oct 31st 2020

## R Markdown

##Read the Binary csv dataset and use summary() on the dataframe.

cluster\_csv <- read.csv('clustering-data.csv', header = TRUE)  
  
str(cluster\_csv)

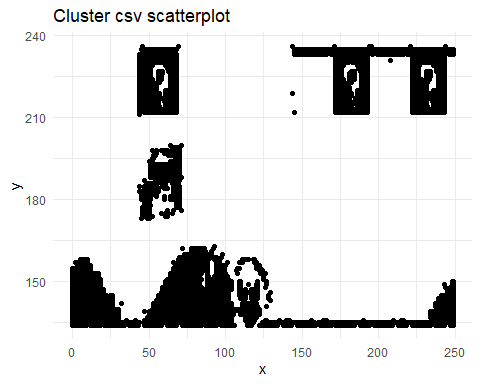
## 'data.frame': 4022 obs. of 2 variables:  
## $ x: int 46 69 144 171 194 195 221 244 45 47 ...  
## $ y: int 236 236 236 236 236 236 236 236 235 235 ...

summary(cluster\_csv)

## 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

**a. Plot scatter plots for dataset**

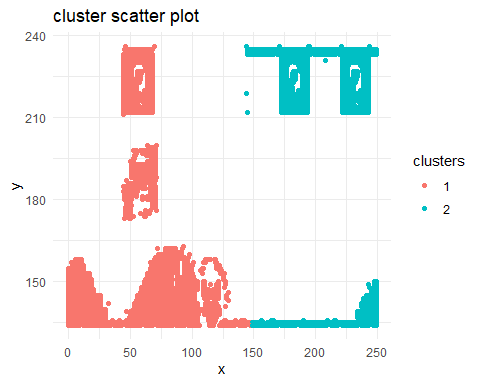
library(ggplot2)  
theme\_set(theme\_minimal())  
  
ggplot(cluster\_csv, aes(x = x, y = y)) + geom\_point() + ggtitle("Cluster csv scatterplot")



## 

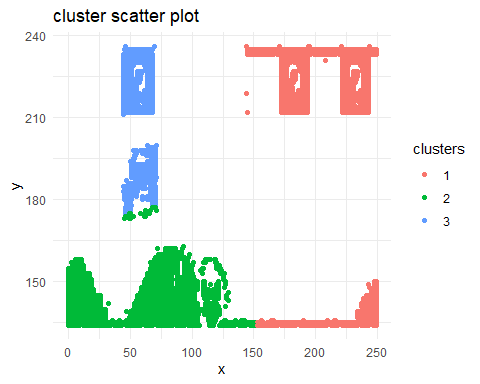
k\_list = c(2:12)  
  
fun\_new <- function(df, n)  
{  
   
 kmeans\_n <- kmeans(df, n)  
 kmeans\_n$cluster <- as.factor(kmeans\_n$cluster)  
 clusters <- kmeans\_n$cluster  
   
 print(sprintf("Scatter plot of cluster with k=%s", n))  
   
 ggplot(df, aes(x, y, col=clusters)) + geom\_point() + ggtitle("cluster scatter plot")  
   
}  
  
   
fun\_new(cluster\_csv, 2)

## [1] "Scatter plot of cluster with k=2"



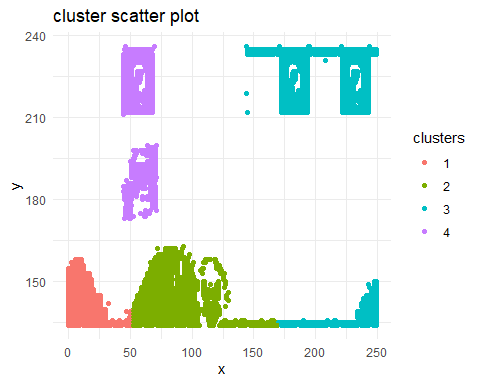
fun\_new(cluster\_csv, 3)

## [1] "Scatter plot of cluster with k=3"



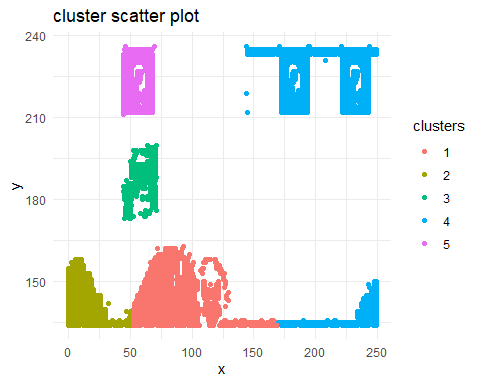
fun\_new(cluster\_csv, 4)

## [1] "Scatter plot of cluster with k=4"



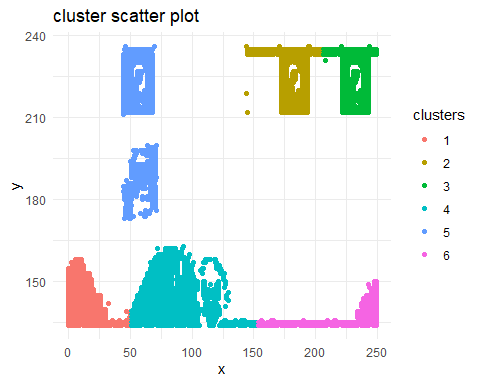
fun\_new(cluster\_csv, 5)

## [1] "Scatter plot of cluster with k=5"



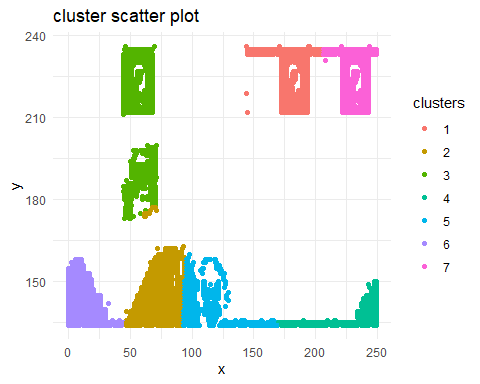
fun\_new(cluster\_csv, 6)

## [1] "Scatter plot of cluster with k=6"



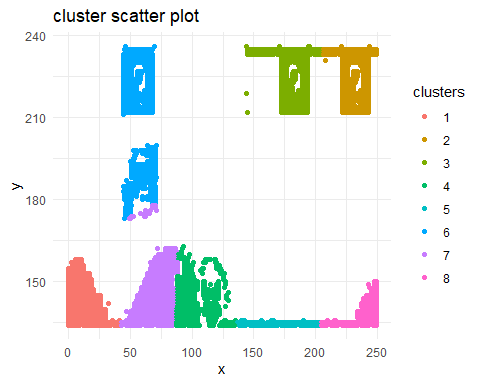
fun\_new(cluster\_csv, 7)

## [1] "Scatter plot of cluster with k=7"



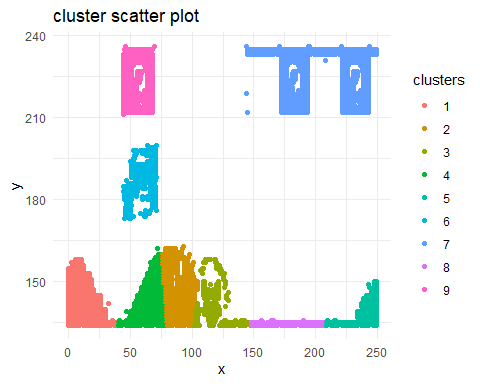
fun\_new(cluster\_csv, 8)

## [1] "Scatter plot of cluster with k=8"



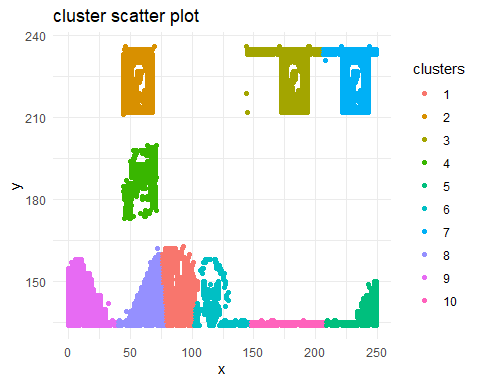
fun\_new(cluster\_csv, 9)

## [1] "Scatter plot of cluster with k=9"



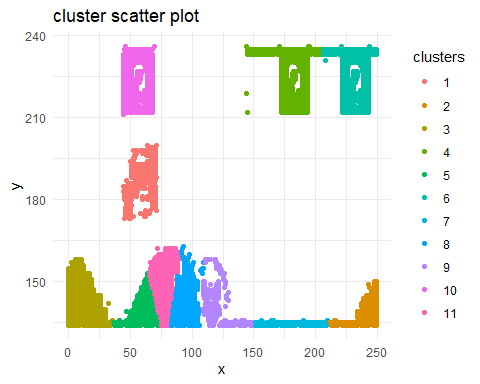
fun\_new(cluster\_csv, 10)

## [1] "Scatter plot of cluster with k=10"



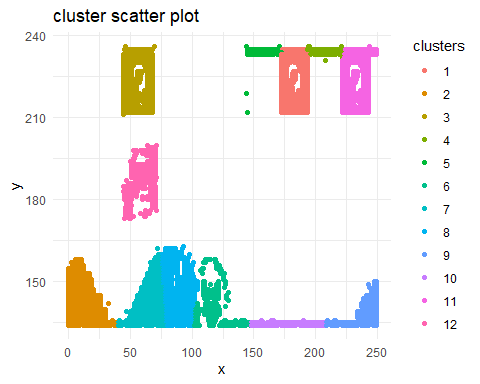
fun\_new(cluster\_csv, 11)

## [1] "Scatter plot of cluster with k=11"



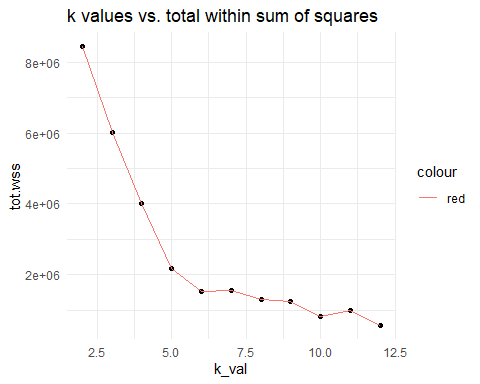
fun\_new(cluster\_csv, 12)

## [1] "Scatter plot of cluster with k=12"



**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.** **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.** **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?**

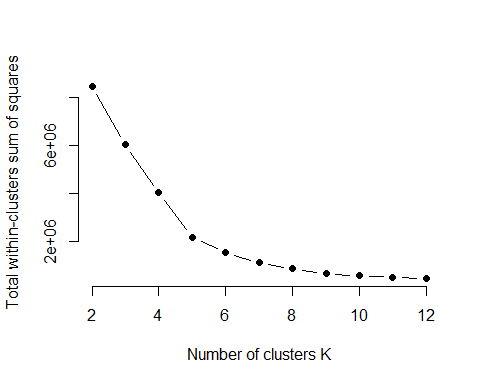
i <- 1  
tmp\_list\_vec <- c()  
  
for(t in k\_list){  
  
 tmp\_list\_vec <- c(tmp\_list\_vec, ((kmeans(cluster\_csv, t))$tot.withinss))  
 i <- i + 1  
  
}  
  
tmp\_list <- data.frame(k\_list, tmp\_list\_vec)  
colnames <- c("k\_val", "tot.wss")  
colnames(tmp\_list) <- colnames  
  
  
ggplot(tmp\_list, aes(k\_val, tot.wss)) + geom\_point() + geom\_line(aes(col="red")) + ggtitle("k values vs. total within sum of squares")



k.max <- 12  
data <- cluster\_csv  
wss <- sapply(2:k.max,   
 function(k){kmeans(data, k, nstart=50,iter.max = 11)$tot.withinss})  
wss

## [1] 8443681.1 6014377.9 4009678.4 2171612.8 1519043.4 1102869.9 853160.1  
## [8] 647331.9 554632.2 499954.4 452352.3

plot(2:k.max, wss,  
 type="b", pch = 19, frame = FALSE,   
 xlab="Number of clusters K",  
 ylab="Total within-clusters sum of squares")



**Answer c.**

The elbow method looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn’t give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the “elbow criterion”. This “elbow” cannot always be unambiguously identified. (r-bloggers.com reference)

Considering above points and Looking at above two plots, attempted couple of ways, we can see that beyond k=5, the marginal gain starts to drop. Hence k=5 would be an optimal value for given dataset. Even, k=7 could be a potential candidate, but beyond this value of k, no significant value can be achieved.

## References

<https://www.r-bloggers.com/2017/02/finding-optimal-number-of-clusters/>

<https://www.analyticsvidhya.com/blog/2015/08/learning-concept-knn-algorithms-programming/#>:~:text=Unhesitatingly%2C%20using%20kNN%20Algorithm.,points%20into%20well%20defined%20groups

<https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/>