**k-Means clustering for Wholesale Data**:

**Code:**

#--------------

# @Authour Nikhil N Pandarge

#--------------

#Loading Datasets:

CustomerData <- read.csv("Wholesale customers data.csv")

Customerdata <- data.frame(CustomerData)

summary(Customerdata)

**Output:**

**> summary(Customerdata)**

**Channel Region Fresh Milk Grocery**

**Min. :1.000 Min. :1.000 Min. : 3 Min. : 55 Min. : 3**

**1st Qu.:1.000 1st Qu.:2.000 1st Qu.: 3128 1st Qu.: 1533 1st Qu.: 2153**

**Median :1.000 Median :3.000 Median : 8504 Median : 3627 Median : 4756**

**Mean :1.323 Mean :2.543 Mean : 12000 Mean : 5796 Mean : 7951**

**3rd Qu.:2.000 3rd Qu.:3.000 3rd Qu.: 16934 3rd Qu.: 7190 3rd Qu.:10656**

**Max. :2.000 Max. :3.000 Max. :112151 Max. :73498 Max. :92780**

**Frozen Detergents\_Paper Delicassen**

**Min. : 25.0 Min. : 3.0 Min. : 3.0**

**1st Qu.: 742.2 1st Qu.: 256.8 1st Qu.: 408.2**

**Median : 1526.0 Median : 816.5 Median : 965.5**

**Mean : 3071.9 Mean : 2881.5 Mean : 1524.9**

**3rd Qu.: 3554.2 3rd Qu.: 3922.0 3rd Qu.: 1820.2**

**Max. :60869.0 Max. :40827.0 Max. :47943.0**

**Code:**

sum(is.na(Customerdata))

**Output:**

**0**

**Code:**

top.n.custs <- function (Customerdata,cols,n=5)

{

#Initialize a vector to hold customers being removed

idx.to.remove <-integer(0)

for (c in cols) # For every column in the data we passed to this function

{

col.order <-order(Customerdata[,c],decreasing=T) #Sort column "c" in descending order (bigger on top)

#Order returns the sorted index (e.g. row 15, 3, 7, 1, ...) rather than the actual values sorted.

idx <-head(col.order, n)#Take the first n of the sorted column C to

idx.to.remove <-union(idx.to.remove,idx)

#combine and de-duplicate the row ids that need to be removed

}

#Return the indexes of customers to be removed

return(idx.to.remove)

}

#How Many Customers to be Removed?

top.custs <-top.n.custs(Customerdata, cols = 1:5,n=5)

length(top.custs)

**Output:**

**[1] 18**

**Code:**

#Examine the customers

Customerdata[top.custs,] #Exammine the customers

Channel Region Fresh Milk Grocery Frozen Detergents\_Paper Delicassen

**Output:**

**> #Examine the customers**

**> Customerdata[top.custs,] #Exammine the customers**

**Channel Region Fresh Milk Grocery Frozen Detergents\_Paper Delicassen**

**1 2 3 12669 9656 7561 214 2674 1338**

**2 2 3 7057 9810 9568 1762 3293 1776**

**3 2 3 6353 8808 7684 2405 3516 7844**

**5 2 3 22615 5410 7198 3915 1777 5185**

**6 2 3 9413 8259 5126 666 1795 1451**

**4 1 3 13265 1196 4221 6404 507 1788**

**182 1 3 112151 29627 18148 16745 4948 8550**

**126 1 3 76237 3473 7102 16538 778 918**

**285 1 3 68951 4411 12609 8692 751 2406**

**40 1 3 56159 555 902 10002 212 2916**

**259 1 1 56083 4563 2124 6422 730 3321**

**87 2 3 22925 73498 32114 987 20070 903**

**48 2 3 44466 54259 55571 7782 24171 6465**

**86 2 3 16117 46197 92780 1026 40827 2944**

**184 1 3 36847 43950 20170 36534 239 47943**

**62 2 3 35942 38369 59598 3254 26701 2017**

**334 2 2 8565 4980 67298 131 38102 1215**

**66 2 3 85 20959 45828 36 24231 1423**

**Code:**

set.seed(76964057) #Set the seed for reproducibility

#Try K from 2 to 20

rng<-2:20

#Number of times to run the K Means algorithm

tries <-100

#Set up an empty vector to hold all of points

avg.totw.ss <-integer(length(rng))

avg.totb.ss <- integer(length(rng))

avg.tot.ss <- integer(length(rng))

# For each value of the range variable

for(v in rng){

#Set up an empty vectors to hold the tries

v.totw.ss <-integer(tries)

b.totb.ss <- integer(tries)

tot.ss <- integer(tries)

#Run kmeans

for(i in 1:tries){

k.temp <-kmeans(data.rm.top,centers=v)

#Store the total withinss

v.totw.ss[i] <-k.temp$tot.withinss

#Store the betweenss

b.totb.ss[i] <- k.temp$betweenss

#Store the total sum of squares

tot.ss[i] <- k.temp$totss

}

#Average the withinss and betweenss

avg.totw.ss[v-1] <-mean(v.totw.ss)

avg.totb.ss[v-1] <-mean(b.totb.ss)

avg.tot.ss[v-1] <- mean(tot.ss)

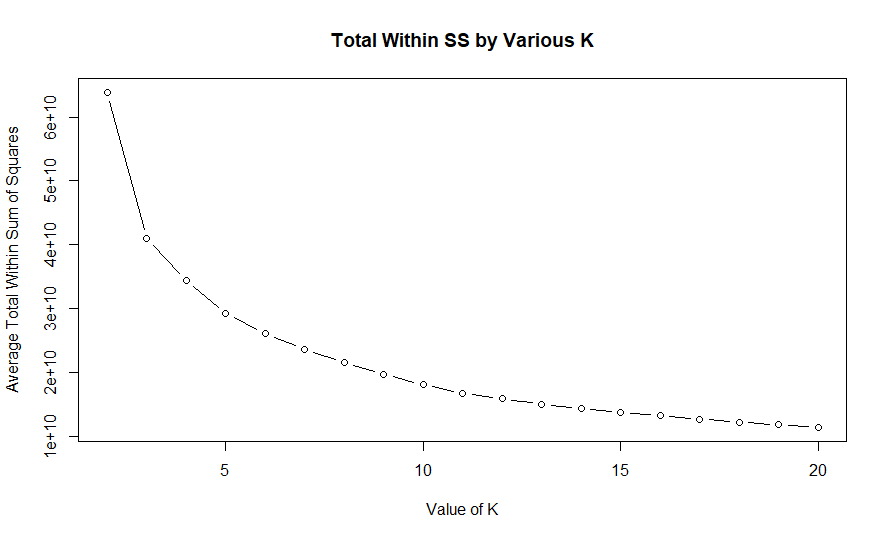
}

plot(rng,avg.totw.ss,type="b", main="Total Within SS by Various K",

ylab="Average Total Within Sum of Squares",

xlab="Value of K")

**Output:**

****

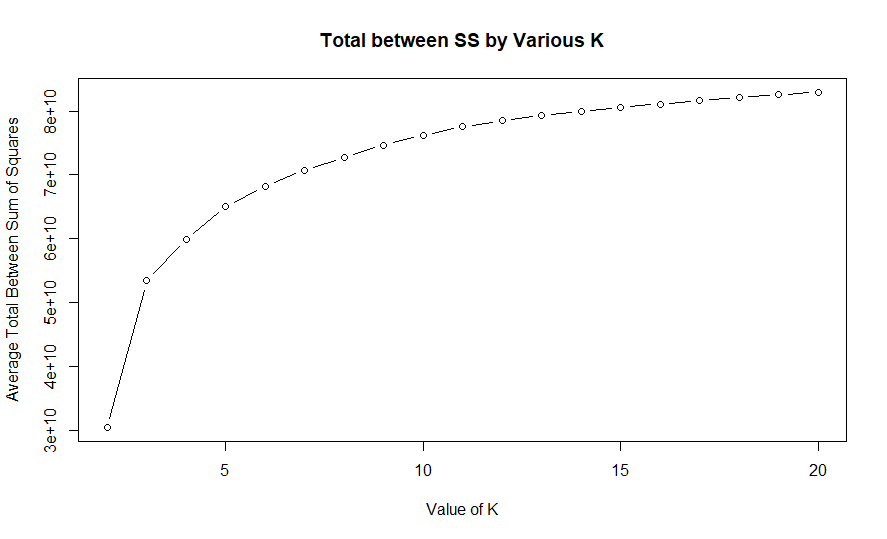
**Code:**

plot(rng,avg.totb.ss,type="b", main="Total between SS by Various K",

ylab="Average Total Between Sum of Squares",

xlab="Value of K")

**Output:**

****

**Code:**

#Plot the ratio of betweenss/total ss and withinss / total ss for evaluation

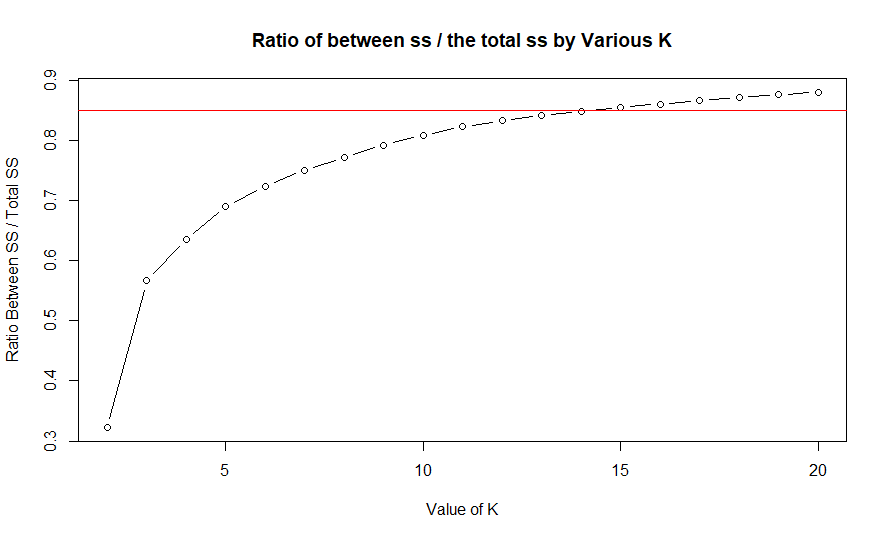
plot(rng,avg.totb.ss/avg.tot.ss,type="b", main="Ratio of between ss / the total ss by Various K",

ylab="Ratio Between SS / Total SS",

xlab="Value of K")

abline(h=0.85, col="red")

**Output:**

****

**Code:**

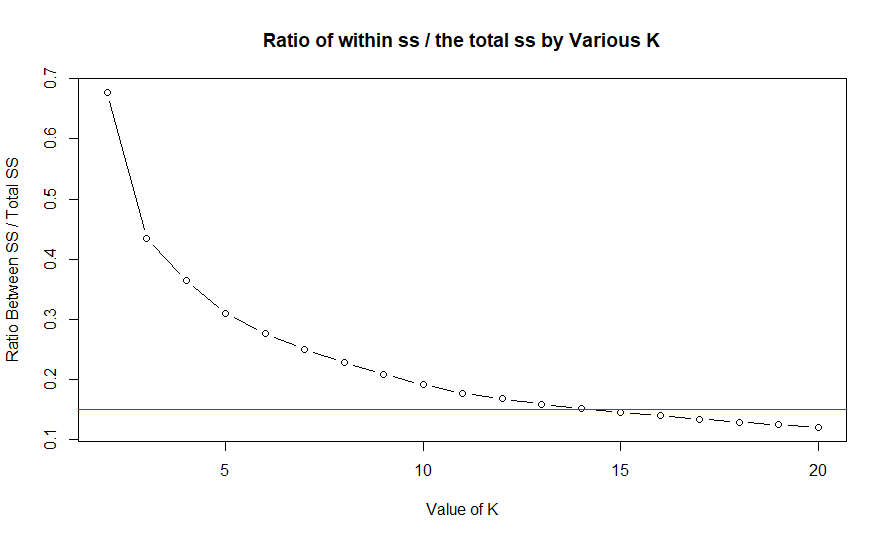
plot(rng,avg.totw.ss/avg.tot.ss,type="b", main="Ratio of within ss / the total ss by Various K",

ylab="Ratio Between SS / Total SS",

xlab="Value of K")

abline(h=0.15, col="red")

**Output:**

****

**Code:**

#Create the best number of clusters, Remove columns 1 and 2

n <- 3

#return(as.integer(n))

k <-kmeans(data.rm.top[,-c(1,2)], centers=n)

print(k$centers)

#Give a count of data points in each cluster

print(table(k$cluster))

#Generate a plot of the clusters

library(cluster)

clusplot(data.rm.top, k$cluster, main='2D representation of the Cluster solution',

color=TRUE, shade=TRUE,

labels=2, lines=0)

**Output:**

**> print(k$centers)**

**Fresh Milk Grocery Frozen Detergents\_Paper Delicassen**

**1 6672.675 3145.713 3989.135 2586.316 1169.093 965.7046**

**2 25984.252 4127.505 5426.165 4675.359 1126.621 1856.1456**

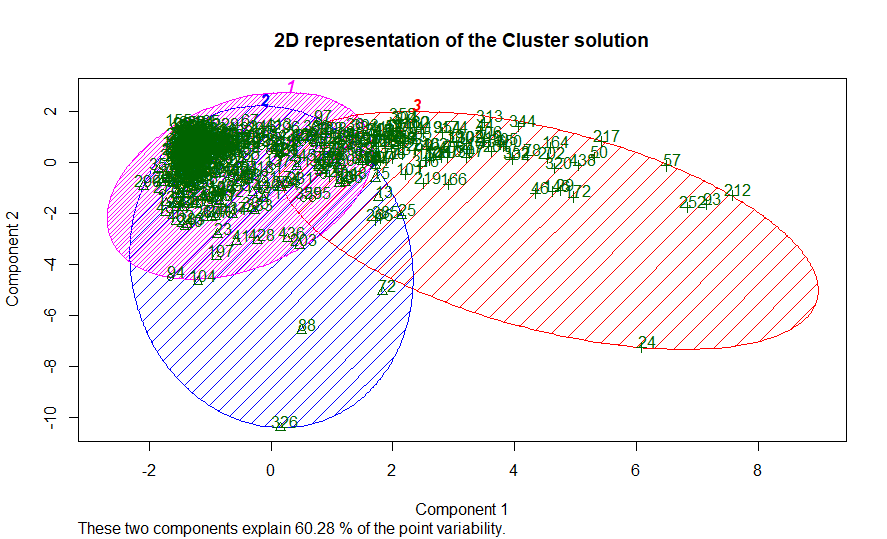
**3 5078.402 12337.927 18763.902 1629.463 8285.585 1835.1829**

**> #Give a count of data points in each cluster**

**> print(table(k$cluster))**

**1 2 3**

**237 103 82**

****