Assignment 3: Orthopedic Materials Sales (100 points)
Student Name: Abdul Rahman
Fall 2019

Purpose: To perform cluster analysis to identify potential business for orthopedic material sales
Description: The objective of this study is to find ways to increase sales of orthopedic material from our company to hospitals in the United States. The data include information about over 4000 hospitals. Below is the data dictionary:

ZIP: US POSTAL CODE
HID: HOSPITAL ID
CITY: CITY NAME
STATE: STATE NAME
BEDS: NUMBER OF HOSPITAL BEDS
RBEDS: NUMBER OF REHAB BEDS
OUT-V: NUMBER OF OUTPATIENT VISITS
ADM: ADMINISTRATIVE COST (In $1000's per year)
SIR: REVENUE FROM INPATIENT
SALESY: SALES OF REHABILITATION EQUIPMENT SINCE JAN 1
SALES12: SALES OF REHAB. EQUIP. FOR THE LAST 12 MO
HIP: NUMBER OF HIP OPERATIONS FOR TWO YEARS AGO
KNEE: NUMBER OF KNEE OPERATIONS FOR TWO YEARS AGO
TH: TEACHING HOSPITAL? 0, 1
TRAUMA: DO THEY HAVE A TRAUMA UNIT? 0, 1
REHAB: DO THEY HAVE A REHAB UNIT? 0, 1
HIP12: NUMBER HIP OPERATIONS FOR THE LAST 12 MO
KNEE12: NUMBER KNEE OPERATIONS FOR THE LAST 12 MO
FEMUR12: NUMBER FEMUR OPERATIONS FOR THE LAST 12 MO

Instructions: Please follow these steps:

1. In Canvas, navigate to Assignments and then Assignment4
2. Download and save the data set hospital_ortho.csv
3. Read the file:

```
data <- fread("hospital_ortho.csv", sep=",", header=T, strip.white = T, na.strings = c("NA","NaN","","?"))
```

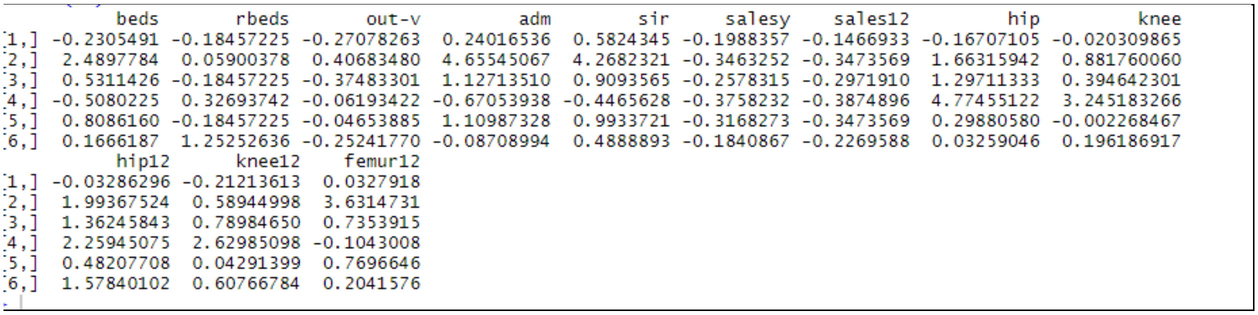
4. The original data includes hospitals across the US. However, we can only sell our products in NC and the nearby states of SC, VA, GA, and TN. Use the following code to narrow down the data to hospitals in these states.

nc_data <- data[(data$state == "NC") | (data$state == "SC") | (data$state == "VA") | (data$state == "GA") | (data$state == "TN")]

4.1. **(3 points)** Look at each individual variable and decide if it should be included in cluster analysis. For those variables that you decide not to include, give your reasons for exclusion.
As clustering is about calculating the distances, the variables which are numeric or continuous are important for analysis. Those values which are categorical and binary need not be included for clustering.
EXCLUDED VARIABLES:

- **[1] ZIP ID and HID: We can exclude these variables as they contain just hospital and its location.**
- **[2] City and State: Since they contain categorical variables so we can exclude them clustering.**
- **[3] TH, Trauma and Rehab: Since they have binary values, we are excluding them too.**

4.2. **(3 points)** Do you need to scale this data? Why?
Yes. We need to scale this data. This would ensure that the differences in scales across variables do not impact the results.

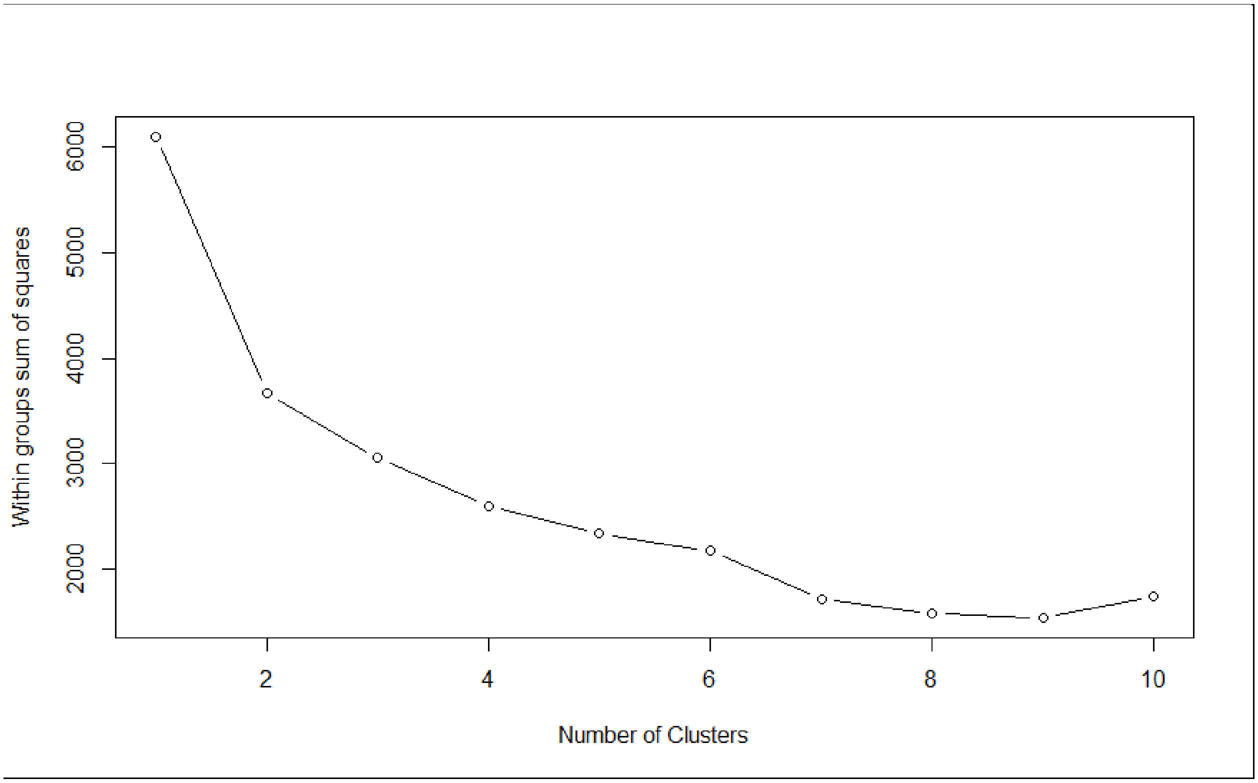


```
         beds       rbeds       out-v         adm        sir     salesy    sales12         hip         knee
[1,] -0.2305491 -0.18457225 -0.27078263  0.24016536  0.5824345 -0.1988357 -0.1466933 -0.16707105 -0.020309865
[2,]  2.4897784  0.05900378  0.40683480  4.65545067  4.2682321 -0.3463252 -0.3473569  1.66315942  0.881760060
[3,]  0.5311426 -0.18457225 -0.37483301  1.12713510  0.9093565 -0.2578315 -0.2971910  1.29711333  0.394642301
[4,] -0.5080225  0.32693742 -0.06193422 -0.67053938 -0.4465628 -0.3758232 -0.3874896  4.77455122  3.245183266
[5,]  0.8086160 -0.18457225 -0.04653885  1.10987328  0.9933721 -0.3168273 -0.3473569  0.29880580 -0.002268467
[6,]  0.1666187  1.25252636 -0.25241770 -0.08708994  0.4888893 -0.1840867 -0.2269588  0.03259046  0.196186917
          hip12      knee12    femur12
[1,] -0.03286296 -0.21213613  0.0327918
[2,]  1.99367524  0.58944998  3.6314731
[3,]  1.36245843  0.78984650  0.7353915
[4,]  2.25945075  2.62985098 -0.1043008
[5,]  0.48207708  0.04291399  0.7696646
[6,]  1.57840102  0.60766784  0.2041576
```

5. Perform k-means clustering:

5.1. **(3 points)** Use “Within Groups SSE” to determine the number of clusters for k-means. How many clusters you would like to create?
I prefer to create 4 clusters.

5.2. **(3 points)** Paste the “Within Groups SSE” plot in the space below:



5.3. **(3 points)** Perform k-means clustering using the number of clusters you recommended in 5.1. How many hospitals fall in each cluster?

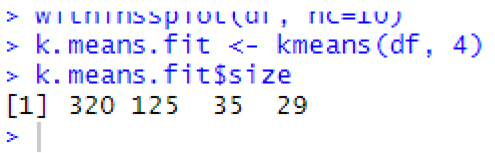
Number of Hospitals in each cluster:

[1] Cluster 1: 320

[2] Cluster 2: 125

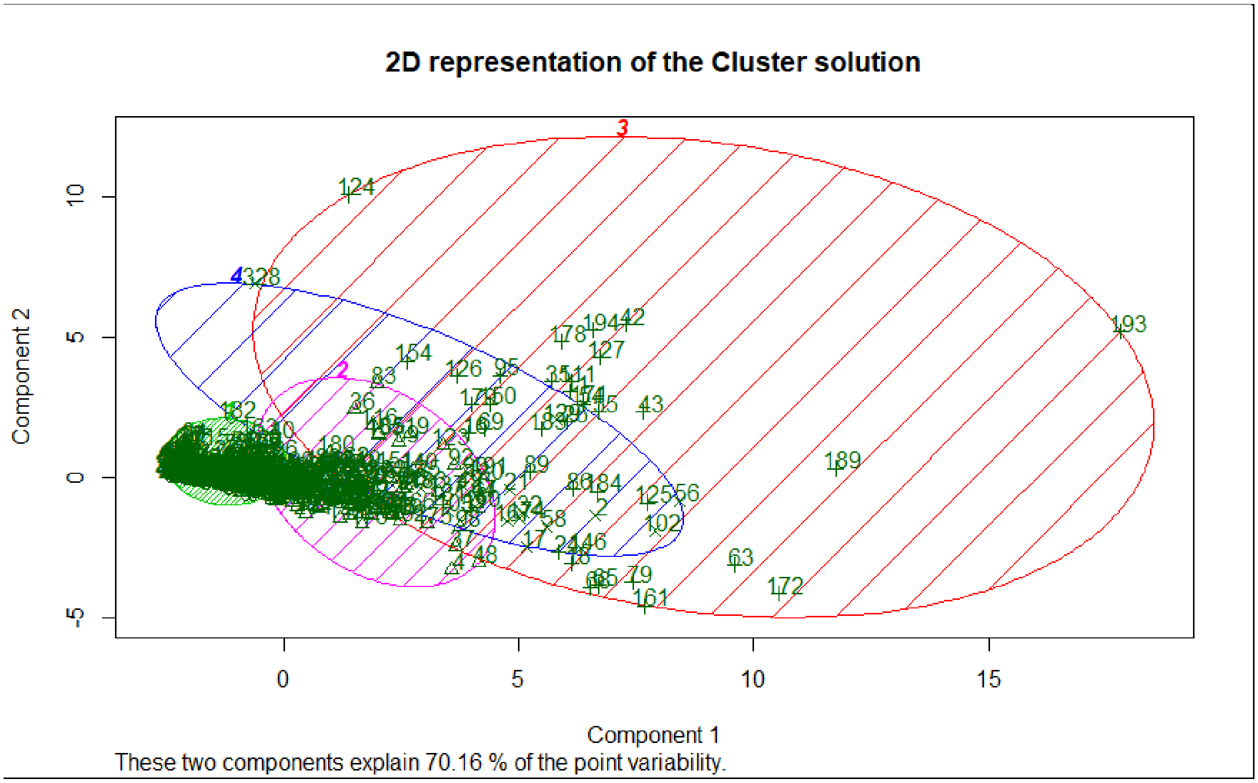
[3] Cluster 3: 35

[4] Cluster 4: 29



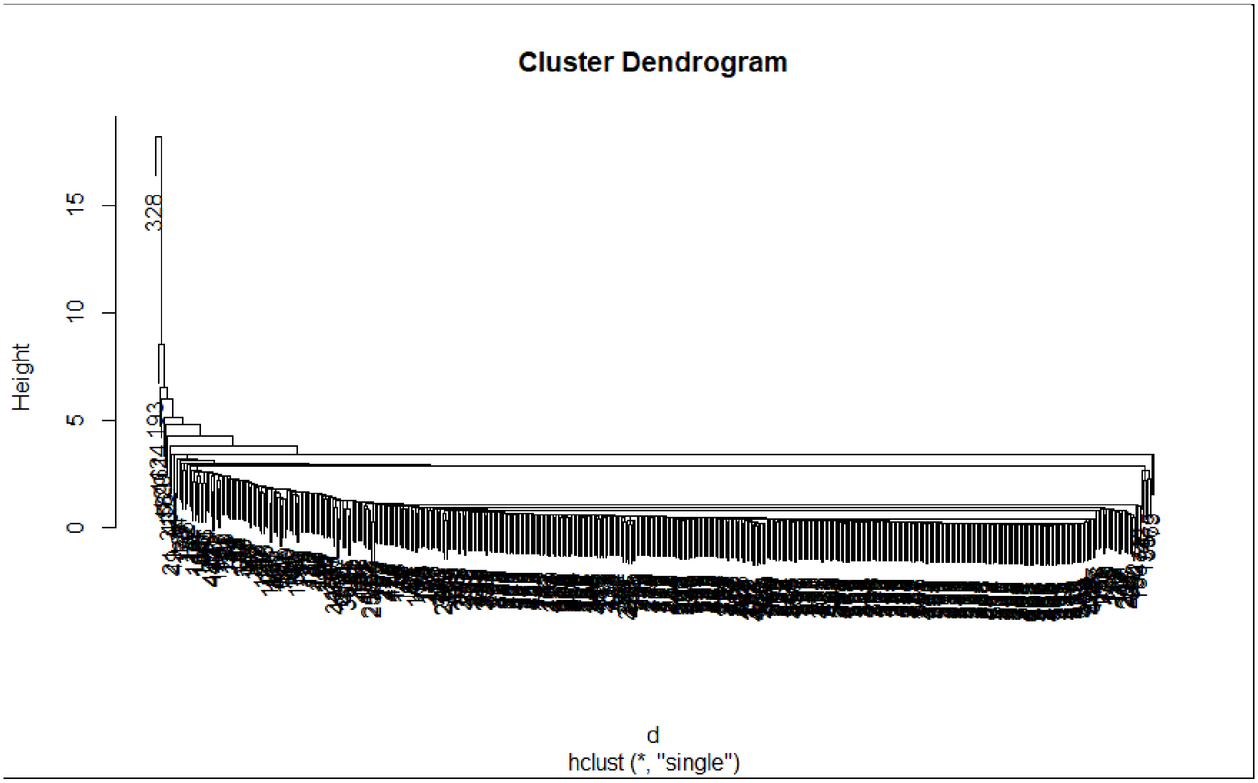
```
> withinssplot(df, nc=10)
> k.means.fit <- kmeans(df, 4)
> k.means.fit$size
[1] 320 125  35  29
> 
```

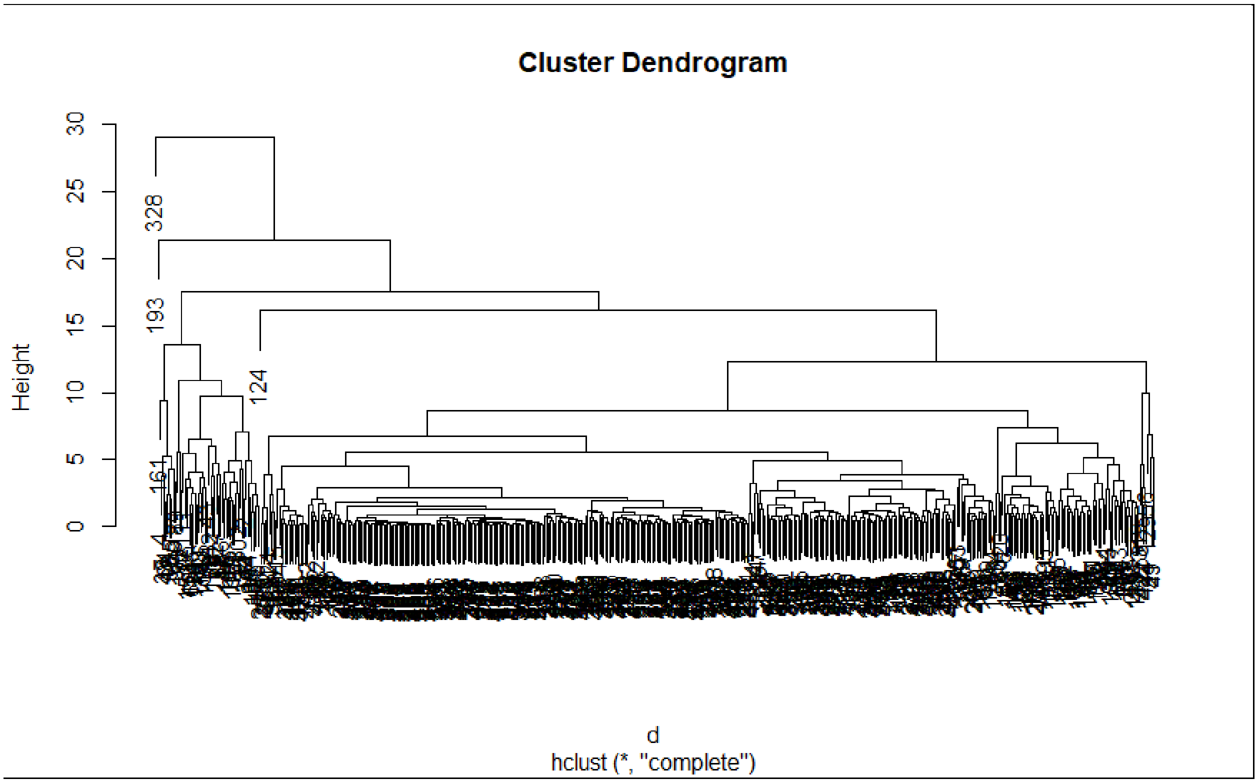
5.4. **(3 points)** Create a two-dimensional representation of the clusters and paste it below:

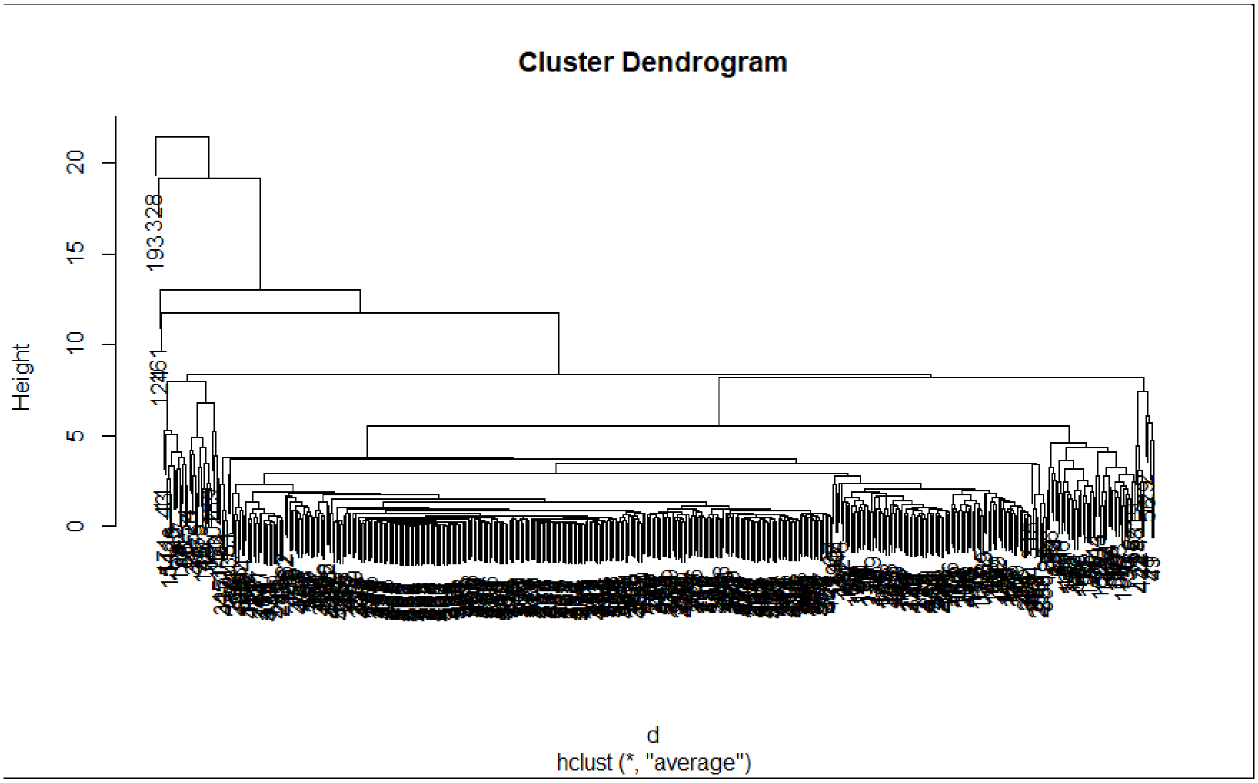


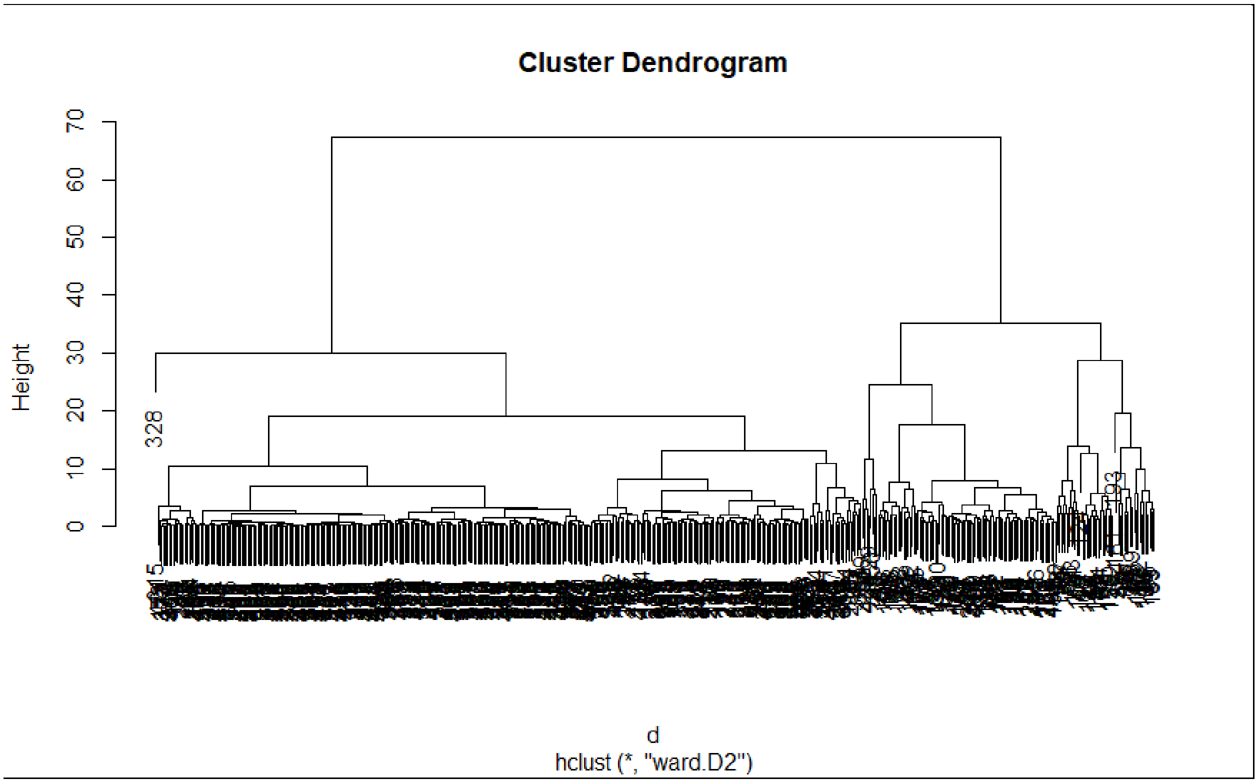
6. Perform Hierarchical clustering.

6.1. **(4 points)** Try different hierarchical clustering and paste the dendrograms in the space below:

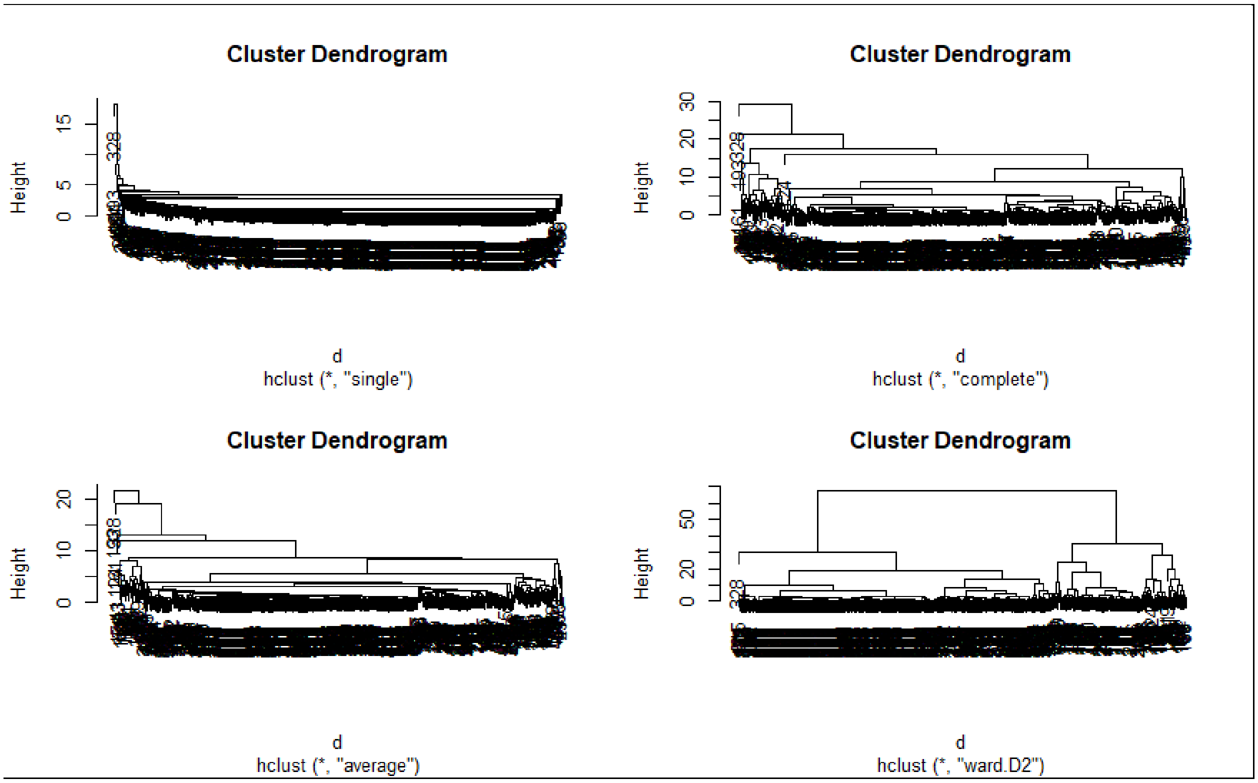

Cluster Dendrogram
Height
0
5
10
15
328
193
124
d
hclust (*, "single")


Cluster Dendrogram
Height
0
5
10
15
20
25
30
328
193
124
161
d
hclust (*, "complete")


Cluster Dendrogram
Height
0
5
10
15
20
193
328
12161
d
hclust (*, "average")


Cluster Dendrogram
Height
0
10
20
30
40
50
60
70
328
d
hclust (*, "ward.D2")

6.2. **(3 points)** Determine which hierarchical clustering method would be more appropriate for this data. Why?



Cluster Dendrogram

Cluster Dendrogram

Cluster Dendrogram

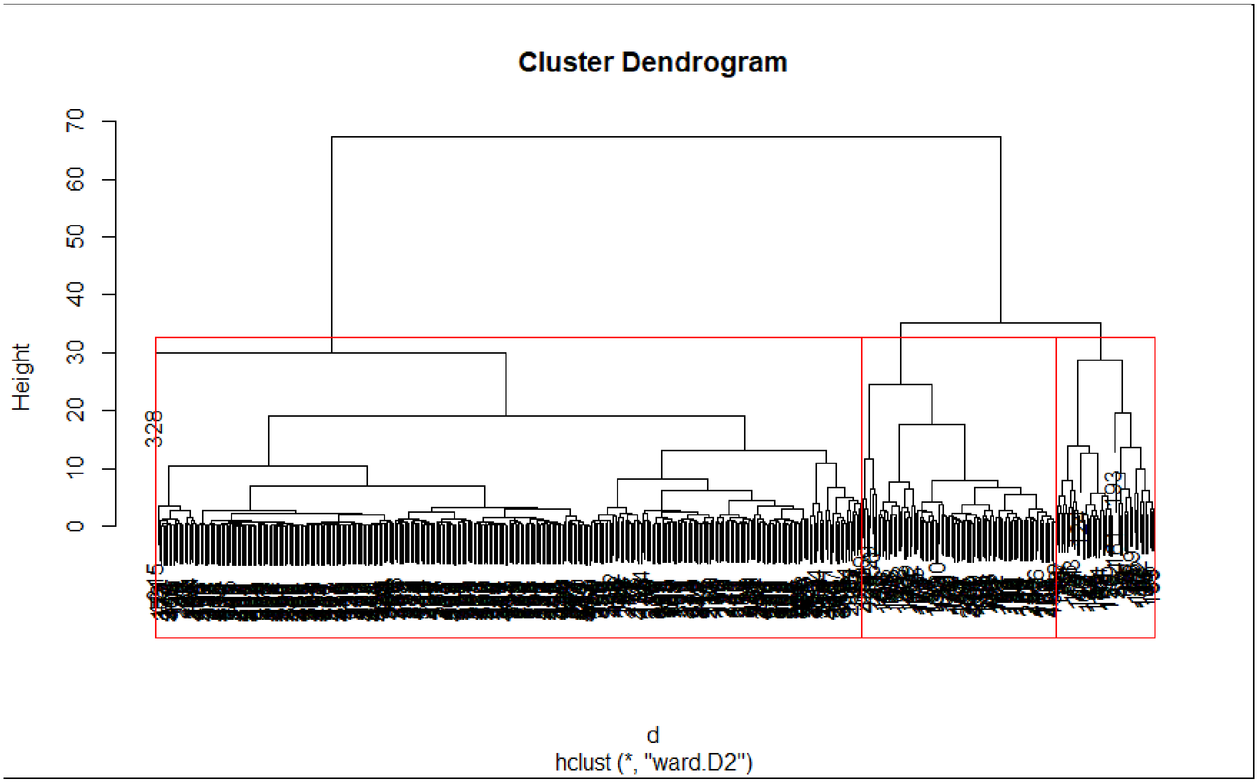
Cluster Dendrogram

I prefer ward method because the datapoints are better distributed in the cluster. Also, the difference between the cluster are well visualized. Finally, Ward method handles outliers in better way than other methods.

6.3. **(3 points)** Based on hierarchical clustering results, how many clusters do you find in this data?

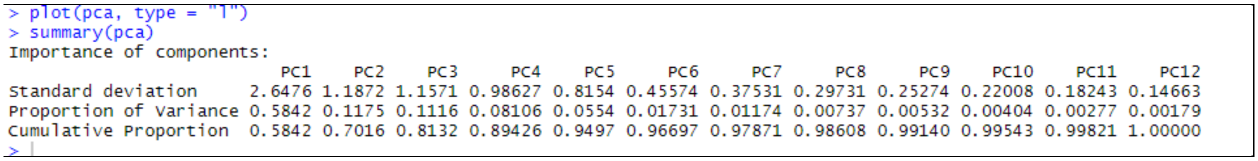
I can find 3 clusters. There is an outlier in the first cluster.

6.4. **(3 points)** Paste the dendrogram that you chose with the red borders around the clusters in the space below:



7. Perform DBSCAN cluster analysis:

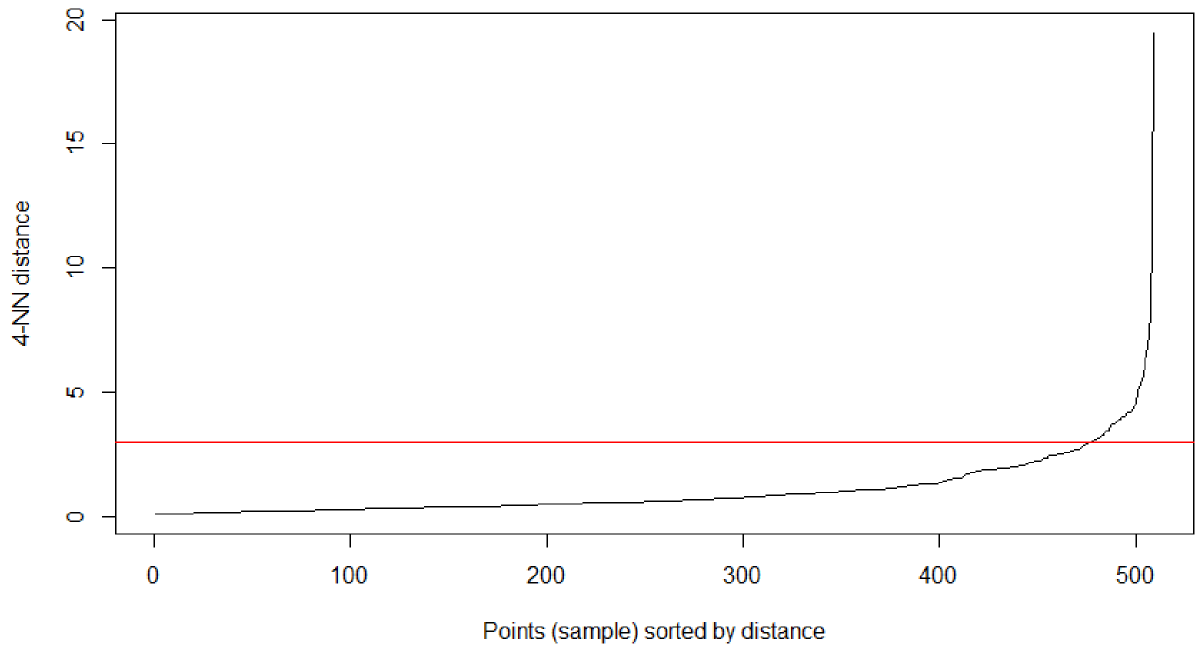
   7.1. (7 points) First, you need to determine minPts. The rule of thumb for minPts is the number of dimensions of the data + 1. Suggest a method to determine the number of dimensions of this data? Implement your method and suggest a good minPts.



```
> plot(pca, type = "l")
> summary(pca)
Importance of components:
                          PC1    PC2    PC3     PC4    PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12
Standard deviation     2.6476 1.1872 1.1571 0.98627 0.8154 0.45574 0.37531 0.29731 0.25274 0.22008 0.18243 0.14663
Proportion of Variance 0.5842 0.1175 0.1116 0.08106 0.0554 0.01731 0.01174 0.00737 0.00532 0.00404 0.00277 0.00179
Cumulative Proportion  0.5842 0.7016 0.8132 0.89426 0.9497 0.96697 0.97871 0.98608 0.99140 0.99543 0.99821 1.00000
> 
```

The number of dimensions of this data can be determined by performing Principal Component analysis. The rule of thumb, to determine the number of PCs for analysis, is by continue adding PCs until there is no significant increase (more than 10%) in the cumulative proportion of variance explained by the PCs. For instance, PC3 adds about 11% to the cumulative proportion of variance and PC4 adds about 8%. Therefore, we can only retain the first three PCs for our analysis. minPts=num of PC's selected+1= 3+1=4

7.2. **(3 points)** Based on your suggested minPts, determine the eps. Explain your recommendation for eps.



I have used KNN distance plot to determine eps. Thus, we can see from the above that eps=3

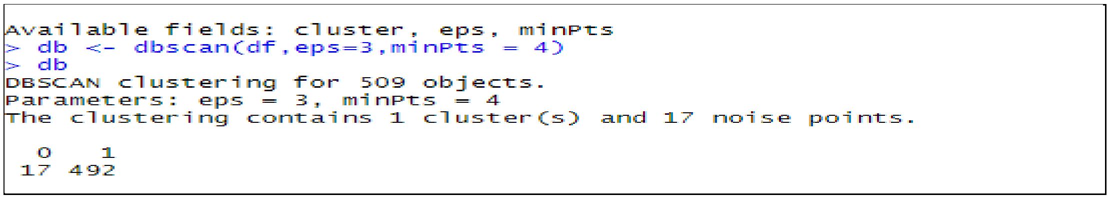
7.3. **(3 points)** Perform DBSCAN clustering using the minPts and eps that you recommended. How many clusters DBSCAN returns?

Parameters: eps = 3, minPts = 4

The clustering contains 1 cluster and 17 noise points.

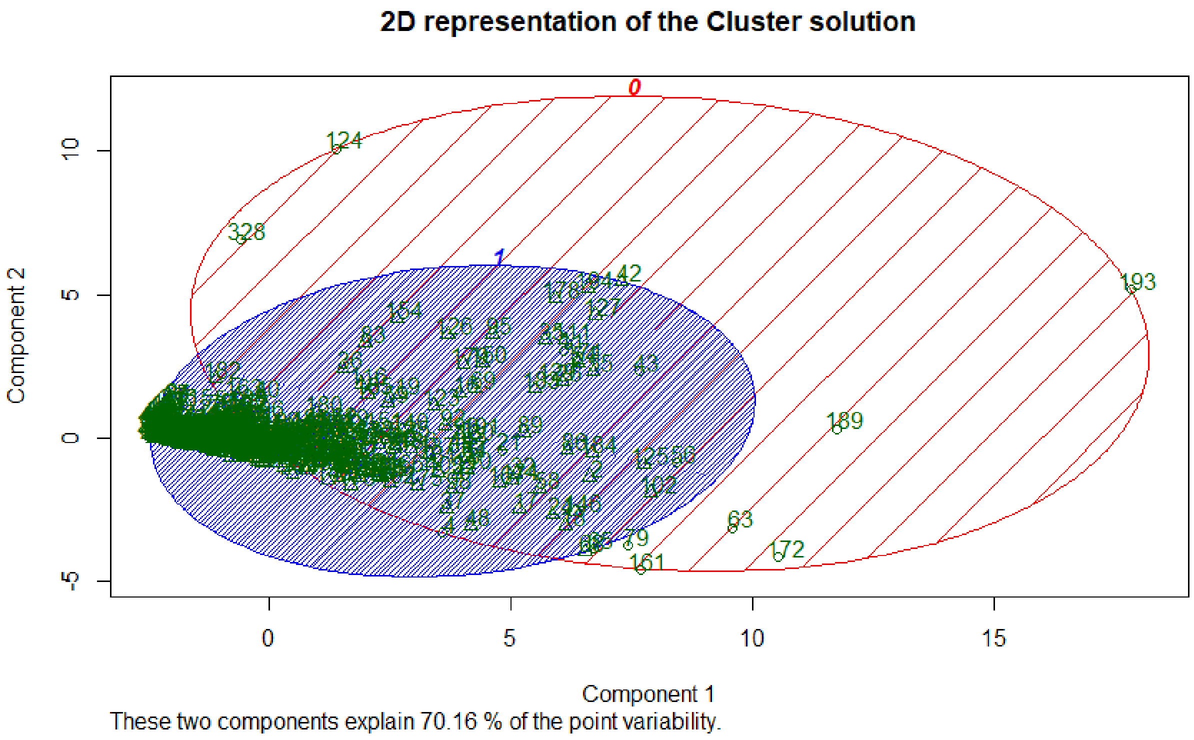
7.4. **(3 points)** How many noise points it returns?

The clustering contains 1 cluster and 17 noise points.



```
Available fields: cluster, eps, minPts
> db <- dbscan(df,eps=3,minPts = 4)
> db
DBSCAN clustering for 509 objects.
Parameters: eps = 3, minPts = 4
The clustering contains 1 cluster(s) and 17 noise points.

  0   1
 17 492
```

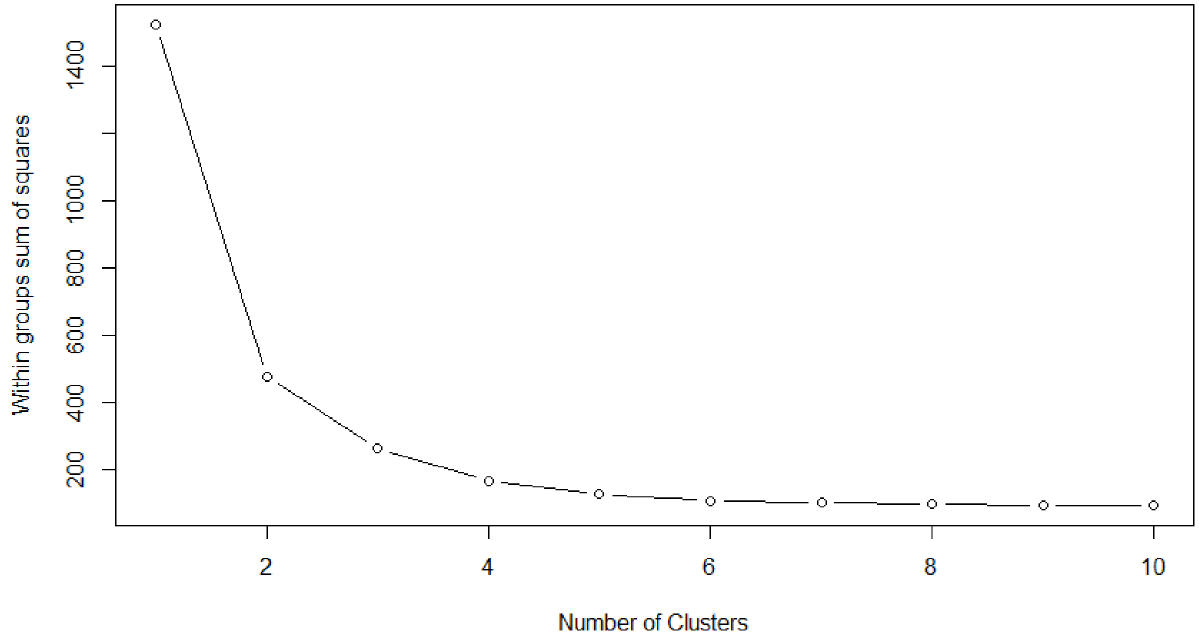
7.5. **(3 points)** Create a two-dimensional representation of DBSCAN cluster(s) and paste it in the space below:

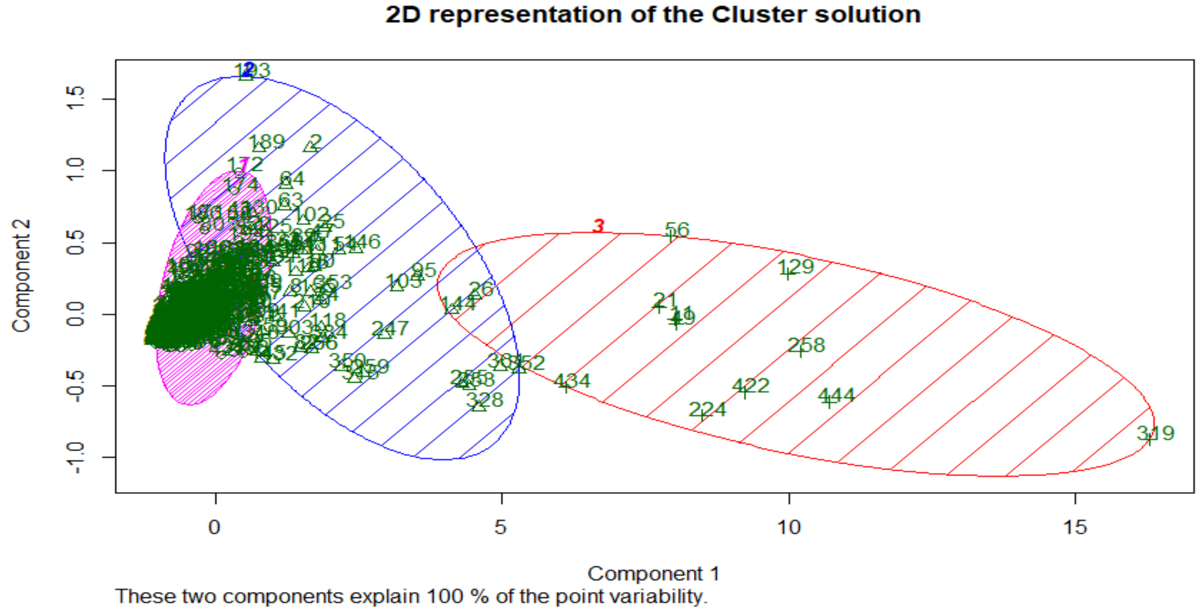


8. Perform principal component analysis on the original data (nc_data). Then select the number of principal components based on PCs variance plot. Let's call the number of PCs n_pc. Then we can use the best PCs instead of the data to perform cluster analysis. To do this, run:

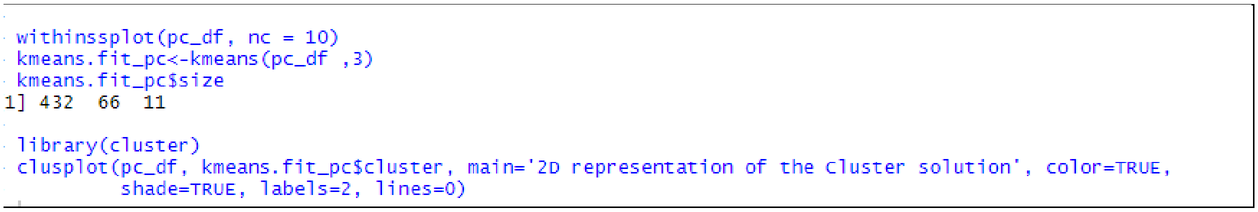
```
pca_data <- predict(pca, newdata = nc_data)
pc_df <- as.data.frame(scale(pca_data[,c(1:n_pc)]))  # replace n_pc with the number of PCs you recommend.
```

8.1. **(10 points)** Repeat your analysis in question 5 using the new pc_df. What is the best k? Paste the two-dimensional representation in the space below:





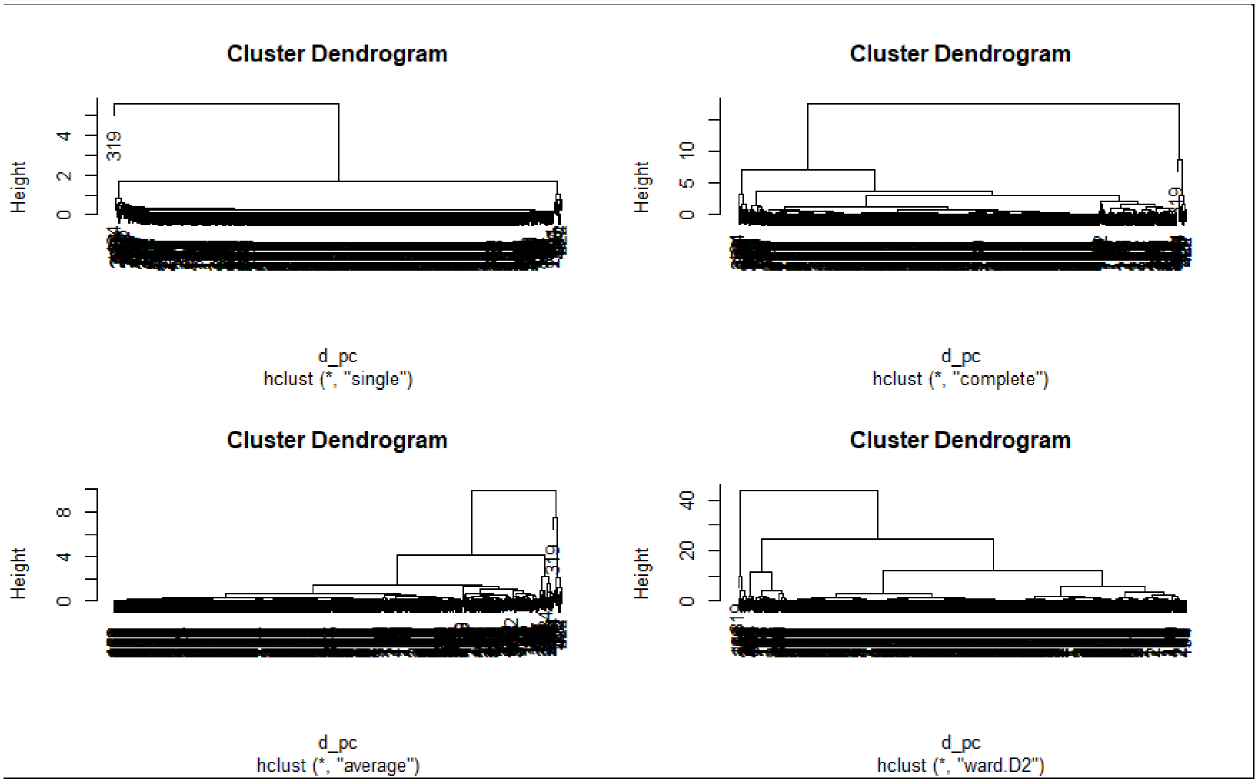
From above, we can see that sum of squares error between clusters converge after 3 clusters. So number of cluster =3. Thus,the best K = 3.



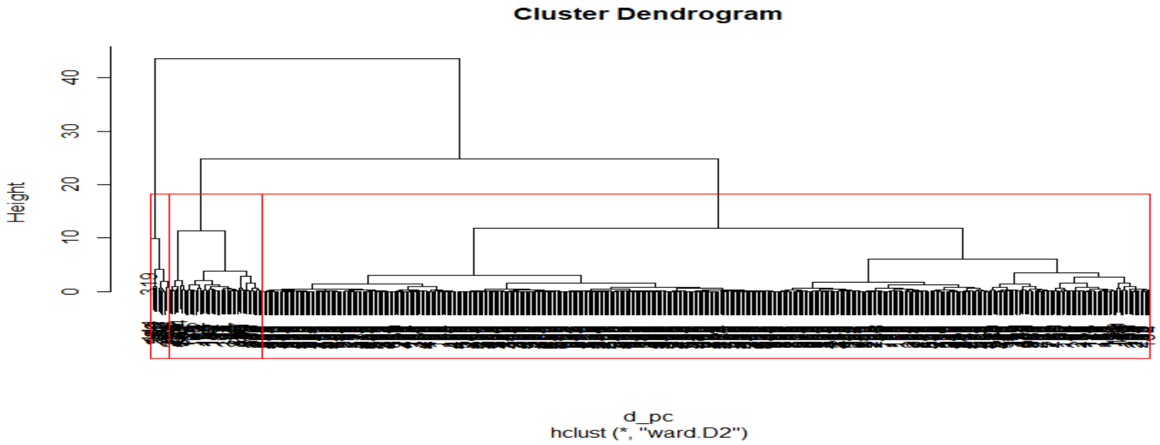
```
withinssplot(pc_df, nc = 10)
kmeans.fit_pc<-kmeans(pc_df ,3)
kmeans.fit_pc$size
1] 432   66   11

library(cluster)
clusplot(pc_df, kmeans.fit_pc$cluster, main='2D representation of the Cluster solution', color=TRUE,
         shade=TRUE, labels=2, lines=0)
```

The number of hospitals in each cluster is : 432, 66 and 11.

8.2. **(10 points)** Repeat your analysis in question 6 using the new pc_df. What is the best method? What is the best k? Paste the dendrogram in the space below:



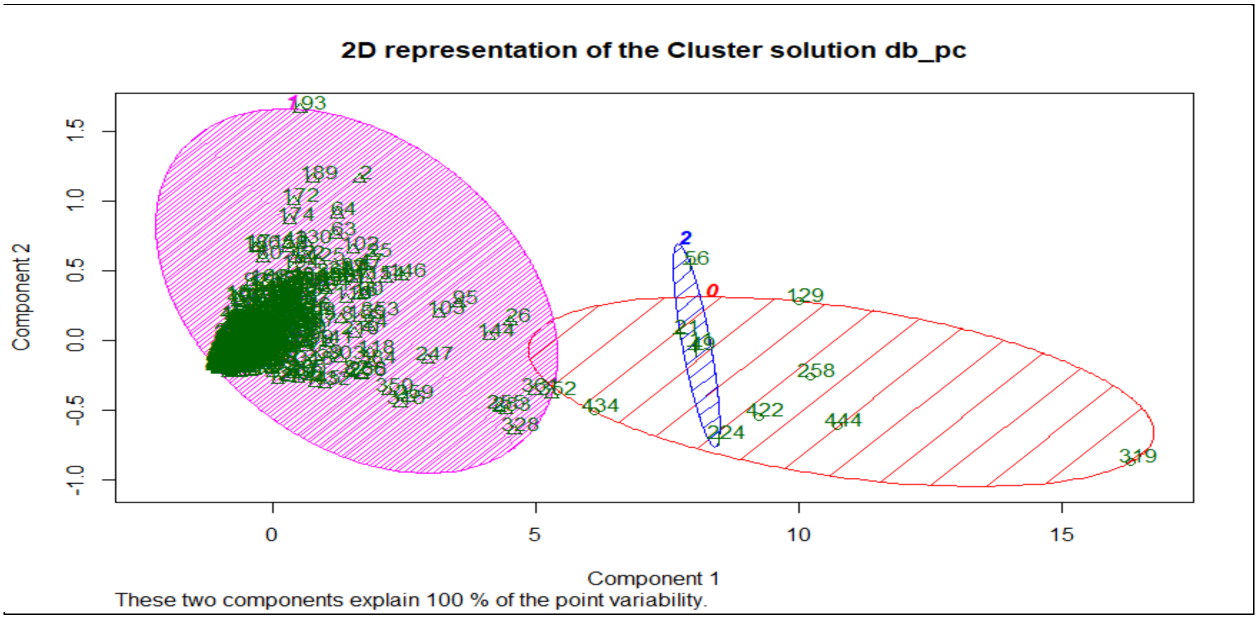
From the above plots we can clearly see that “ward.D2” clustering method works well for the new pc_df data frame. We can see the cluster clearly and Ward.D2 handles the outliers properly. Number of cluster is K = 3.



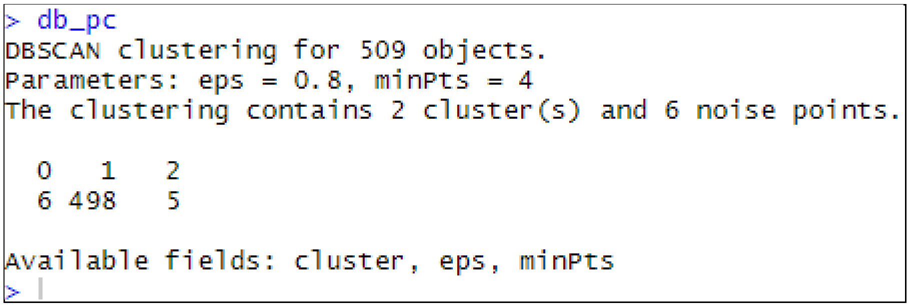
Number of Clusters(K) = 3

8.3. **(10 points)** Repeat your analysis in question 7 using the new pc_df. What is the best minPts? What is the best eps? How many clusters DBSCAN returns? Perform the DBSCAN clustering and paste the two-dimensional representation in the space below:

Number of dimensions in pc_df =3 (Total number of PC's chosen based on Variance plot)
minPts=num of dimensions +1 = 3+1=4



These two components explain 100 % of the point variability.

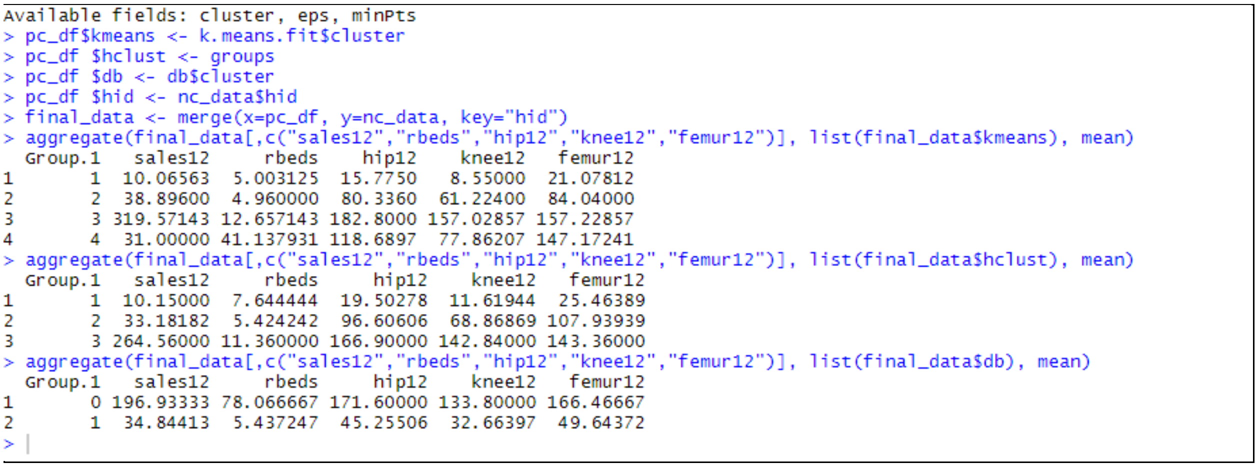


```
> db_pc
DBSCAN clustering for 509 objects.
Parameters: eps = 0.8, minPts = 4
The clustering contains 2 cluster(s) and 6 noise points.

  0   1   2 
  6 498   5 

Available fields: cluster, eps, minPts
> 
```

9. For each hospital, determine the cluster (based on pc_df) to which they belong. Then determine the value of "sales12","rbeds","hip12","knee12", and "femur12" for each cluster for each clustering method (e.g. k-means, hierarchical, DBSCAN). To do this, you need to run the following lines:

pc_df$kmeans <- k.means.fit$cluster
pc_df $hclust <- groups # these groups are created in hierarchical clustering
pc_df $db <- db$cluster
pc_df $hid <- nc_data$hid # Add hospital id to pc_df data
final_data <- merge(x=pc_df, y=nc_data, key="hid")
aggregate(final_data[,c("sales12","rbeds","hip12","knee12","femur12")], list(final_data$kmeans), mean)
aggregate(final_data[,c("sales12","rbeds","hip12","knee12","femur12")], list(final_data$hclust), mean)
aggregate(final_data[,c("sales12","rbeds","hip12","knee12","femur12")], list(final_data$db), mean)

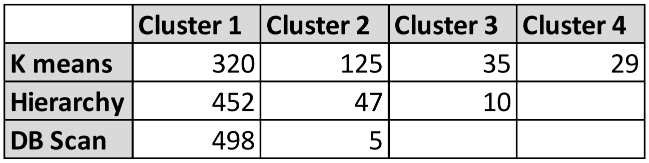
9.1. **(20 points)** Based on these results for each clustering method (e.g. k-means, hierartchical, and DBSCAN), recommend which cluster we should immediately reach out to. Give your reasons.



```
Available fields: cluster, eps, minPts
> pc_df$kmeans <- k.means.fit$cluster
> pc_df $hclust <- groups
> pc_df $db <- db$cluster
> pc_df $hid <- nc_data$hid
> final_data <- merge(x=pc_df, y=nc_data, key="hid")
> aggregate(final_data[,c("sales12","rbeds","hip12","knee12","femur12")], list(final_data$kmeans), mean)
  Group.1   sales12     rbeds    hip12    knee12   femur12
1       1  10.06563  5.003125  15.7750   8.55000  21.07812
2       2  38.89600  4.960000  80.3360  61.22400  84.04000
3       3 319.57143 12.657143 182.8000 157.02857 157.22857
4       4  31.00000 41.137931 118.6897  77.86207 147.17241
> aggregate(final_data[,c("sales12","rbeds","hip12","knee12","femur12")], list(final_data$hclust), mean)
  Group.1   sales12     rbeds     hip12    knee12   femur12
1       1  10.15000  7.644444  19.50278  11.61944  25.46389
2       2  33.18182  5.424242  96.60606  68.86869 107.93939
3       3 264.56000 11.360000 166.90000 142.84000 143.36000
> aggregate(final_data[,c("sales12","rbeds","hip12","knee12","femur12")], list(final_data$db), mean)
  Group.1   sales12     rbeds     hip12    knee12   femur12
1       0 196.93333 78.066667 171.60000 133.80000 166.46667
2       1  34.84413  5.437247  45.25506  32.66397  49.64372
> |
```

Our objective in this assignment is to find ways to increase the sales of orthopedic material to different hospitals in United States. Thus, we have to identify the cluster which have highest

number of hospitals and have larger mean variables for rehab beds, hip operations, knee operations and femur operations facilities. Thus, we can find the below clusters:



	Cluster 1	Cluster 2	Cluster 3	Cluster 4
K means	320	125	35	29
Hierarchy	452	47	10	
DB Scan	498	5		

From above analysis, we take the below parameters for the cluster's having highest hospitals. Thus,

For K means cluster we take Cluster 1 (320) and its mean for sales 12, rbeds12, hip12, knee12 and femur 12 are: 10.05, 5.00,15.77,8.55,21.07

For Hierarchy cluster, we take Cluster 1 (452) and its mean for sales 12, rbeds12, hip12, knee12 and femur 12 are: 10.00,7.64,19.50,11.61,25.4.

For DBSCAN, we take Cluster 1 (498) and its mean for sales 12, rbeds12, hip12, knee12 and femur 12 are :34.8, 5.4, 45.2, 32.6, 49.6

From above we can find that highest mean for the variables sales12, rbeds , hip12, knee12 , femur12 is for DBSCAN.

Also, the mean variables for DBSCAN Cluster 1 is greater than mean variables for Cluster 1 of K means and Hierarchy method, comparing with number of hospitals and sales in respective cluster. i.e. it reflects it have greater potential for sales.

Thus, from the above analysis, we can clearly observe that Cluster 1 from DBSCAN has highest value for each of the variables. Thus, we can target cluster 1 from DBSCAN to increase the sales of the company.

RCODE

```
getwd()
setwd("C:/Users/Abdul Rahman/Documents")
library(readr)
hospital_ortho <- read_csv("hospital_ortho.csv")
View(hospital_ortho)
library(data.table)
library(cluster)
data <- fread("hospital_ortho.csv", sep=",", header=T, strip.white = T, na.strings =
c("NA","NaN","","?"))
head(data,n=20)
nc_data <- data[(data$state == "NC") | (data$state == "SC") | (data$state == "VA") |
(data$state == "GA") | (data$state == "TN")]
head(nc_data,n=30)
df <- scale(nc_data[,c(5:13,17:19)])
head(df)
k.means.fit <- kmeans(df, 3) # Perform k-means clustering with 3 clusters
attributes(k.means.fit)##Check the attributes that k-means generates
k.means.fit$centers # The locations of the centroids
k.means.fit$cluster
k.means.fit$size
library(clustertend)
hopkins(df, n = nrow(df)-1)
withinssplot <- function(data, nc=15, seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares")}
withinssplot(df, nc=10)

k.means.fit <- kmeans(df, 4)
k.means.fit$size
```

```
clusplot(df, k.means.fit$cluster, main='2D representation of the Cluster solution',
color=TRUE, shade=TRUE, labels=2, lines=0)

d <- dist(df, method = "euclidean")
H.single <- hclust(d, method="single")
plot(H.single)

H.complete <- hclust(d, method="complete")
plot(H.complete)

H.average <- hclust(d, method="average")
plot(H.average)
H.ward <- hclust(d, method="ward.D2")
plot(H.ward)


par(mfrow=c(2,2))
plot(H.single)
plot(H.complete)
plot(H.average)
plot(H.ward)

groups <- cutree(H.ward, k=4)
plot(H.ward)
rect.hclust(H.ward, k=4, border="red")
clusplot(df, groups, main='2D representation of the Cluster solution',
         color=TRUE, shade=TRUE,
         labels=2, lines=0)
groups <- cutree(H.ward, k=6)
plot(H.ward)


groups <- cutree(H.ward, k=3)
plot(H.ward)
rect.hclust(H.ward, k=3, border="red")
groups <- cutree(H.ward, k=3)
plot(H.ward)
rect.hclust(H.ward, k=3, border="red")
clusplot(df, groups, main='2D representation of the Cluster solution',
         color=TRUE, shade=TRUE,
        labels=2, lines=0)
pca <-prcomp(df,center=TRUE,scale. = TRUE)
print(pca)
plot(pca, type = "l")
summary(pca)

library(dbscan)
kNNdistplot(df, k =3)
abline(h=3.0, col="red")
db <- dbscan(df,eps=3,minPts = 3)
db

clusplot(df, db$cluster, main='2D representation of the Cluster solution',
        color=TRUE, shade=TRUE,
        labels=2, lines=0)

pca_data <- predict(pca, newdata = nc_data)
pc_df <- as.data.frame(scale(pca_data[,c(1:3)]))


withinssplot(pc_df, nc = 10)
kmeans.fit_pc<-kmeans(pc_df ,3)
kmeans.fit_pc$size

library(cluster)
clusplot(pc_df, kmeans.fit_pc$cluster, main='2D representation of the Cluster solution',
color=TRUE,
        shade=TRUE, labels=2, lines=0)



d_pc <- dist(pc_df, method = "euclidean")
H.single_pc <- hclust(d_pc , method="single")
plot(H.single_pc)
H.complete_pc <- hclust(d_pc , method="complete")
plot(H.complete_pc)
H.average_pc<- hclust(d_pc , method="average")
plot(H.average_pc)
H.ward_pc <- hclust(d_pc , method="ward.D2")
plot(H.ward_pc)
rect.hclust(H.ward_pc, k=3, border="red")
```

```
par(mfrow=c(2,2))
plot(H.single_pc)
plot(H.complete_pc)
plot(H.average_pc)
plot(H.ward_pc)
par(mfrow=c(1,1))
plot(H.ward_pc)
rect.hclust(H.ward_pc, k=3, border="red")
groups_pc <- cutree(H.ward_pc, k=3)
table(groups_pc)

clusplot(pc_df, groups_pc, main="2D representation of the Cluster solution H ward" ,
         color=TRUE, shade=TRUE,
         labels=2, lines=0)

library(dbscan)
kNNdistplot(pc_df, k =3)
abline(h=0.8, col="red")
db_pc <- dbscan(pc_df, eps=0.8, minPts=4)
db_pc$cluster
clusplot(pc_df, db_pc$cluster, main="2D representation of the Cluster solution db_pc",
         color=TRUE, shade=TRUE,
         labels=2, lines=0)




pca <-prcomp(pc_df,center=TRUE,scale. = TRUE)
print(pca)
plot(pca, type = "l")
summary(pca)

library(dbscan)
kNNdistplot(pc_df, k =3)
abline(h=0.8, col="red")
db_pc <- dbscan(pc_df, eps=0.8, minPts=4)
db_pc$cluster
clusplot(pc_df, db_pc$cluster, main="2D representation of the Cluster solution db_pc",
         color=TRUE, shade=TRUE,
         labels=2, lines=0)
db_pc
```

```
pc_df$kmeans <- k.means.fit$cluster
pc_df $hclust <- groups # these groups are created in hierarchical clustering
pc_df $db <- db$cluster
pc_df $hid <- nc_data$hid # Add hospital id to pc_df data
final_data <- merge(x=pc_df, y=nc_data, key="hid")
aggregate(final_data[,c("sales12","rbeds","hip12","knee12","femur12")],
list(final_data$kmeans), mean)
aggregate(final_data[,c("sales12","rbeds","hip12","knee12","femur12")],
list(final_data$hclust), mean)
aggregate(final_data[,c("sales12","rbeds","hip12","knee12","femur12")], list(final_data$db),
mean)
```