

Tarea 4

Wilmer Gonzalez

19 de junio de 2015

Contents

1	Presentación del problema	1
2	Descripción del set de datos	1
3	Respuestas	2

1 Presentación del problema

Responder todas las preguntas presentadas por *Abastos Crema* usando los métodos *hcluster* o *kmeans*

2 Descripción del set de datos

Muestras de laboratorio provistas por el cliente.

```
#setwd("C:\\Users\\isis\\Documents\\DataMining\\clusterizacion_jerarquica\\tarea")

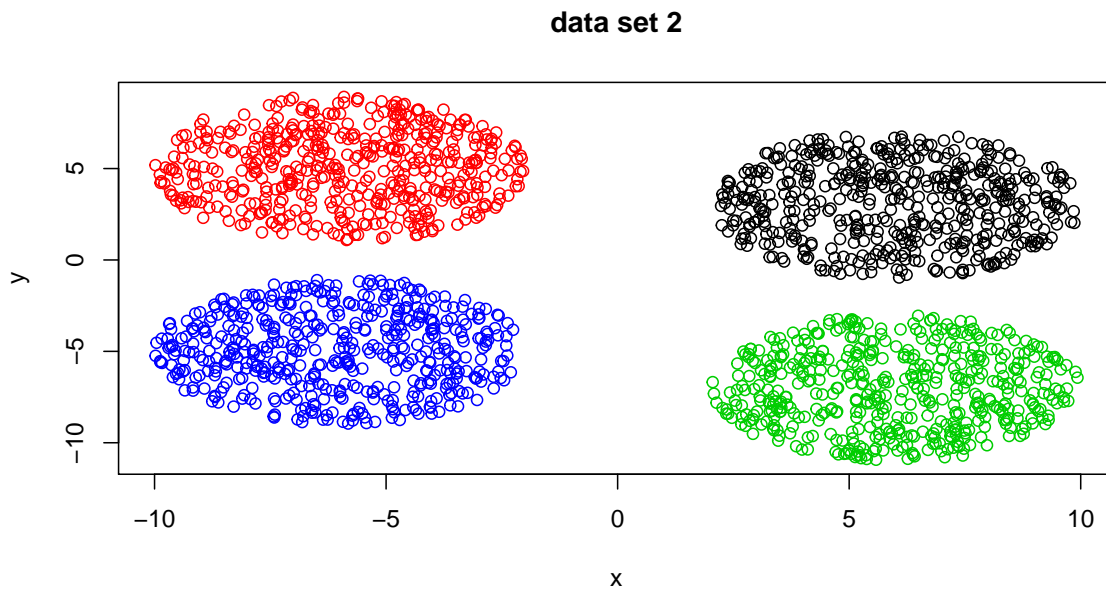
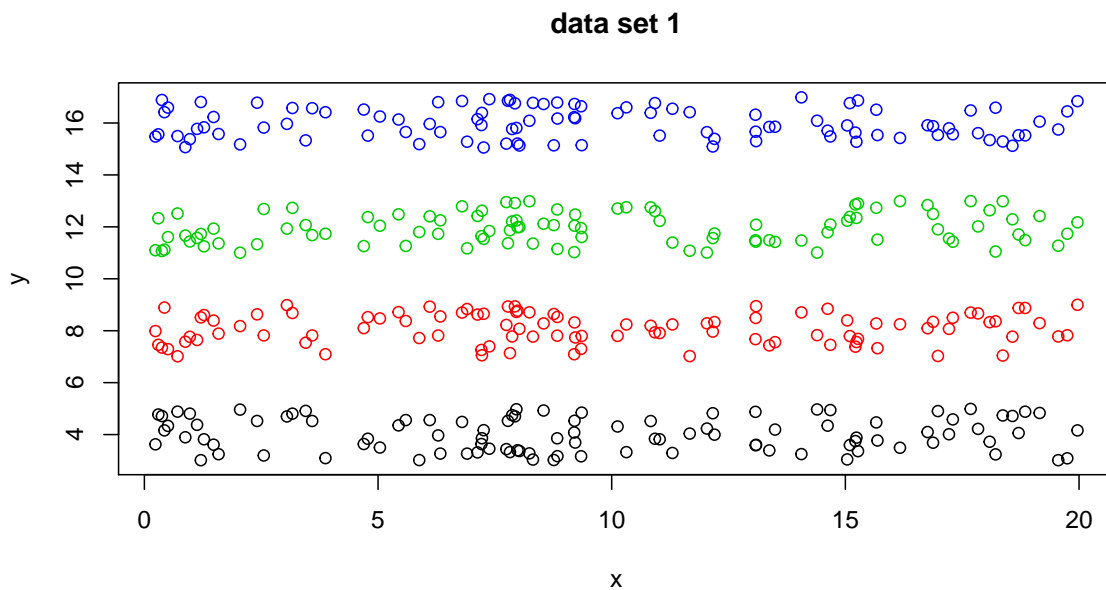
data1 <- read.csv(file= "entrada_1.csv",header = T,sep = ",",dec = ".")
data2 <- read.csv(file= "entrada_2.csv",header = T,sep = ",",dec = ".")
data3 <- read.csv(file= "entrada_3.csv",header = T,sep = ",",dec = ".")
data4 <- read.csv(file= "entrada_4.csv",header = T,sep = ",",dec = ".")
names(data1)<- c("index","x","y","class")
names(data2)<- c("index","x","y","class")
names(data3)<- c("index","x","y","class")
names(data4)<- c("index","x","y","class")
id1 <- rep(1,nrow(data1))
id2 <- rep(2,nrow(data2))
id3 <- rep(3,nrow(data3))
id4 <- rep(4,nrow(data4))

id<- c(id1,id2,id3,id4)
names(id) <- "id"
data <- rbind(data1,data2,data3,data4)
data <- cbind(id,data)
```

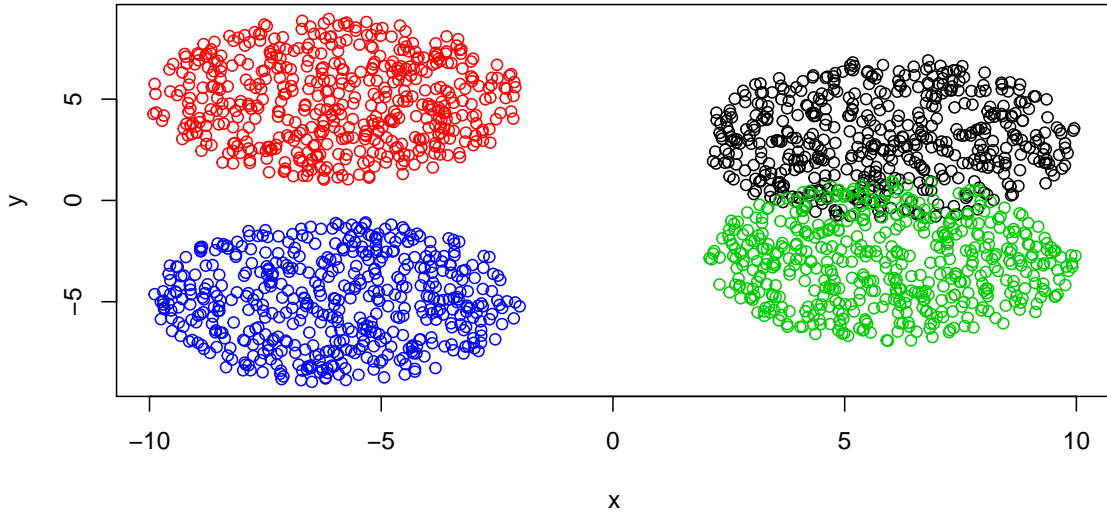
3 Respuestas

1. Grafica de los puntos contenidos en cada set de datos:

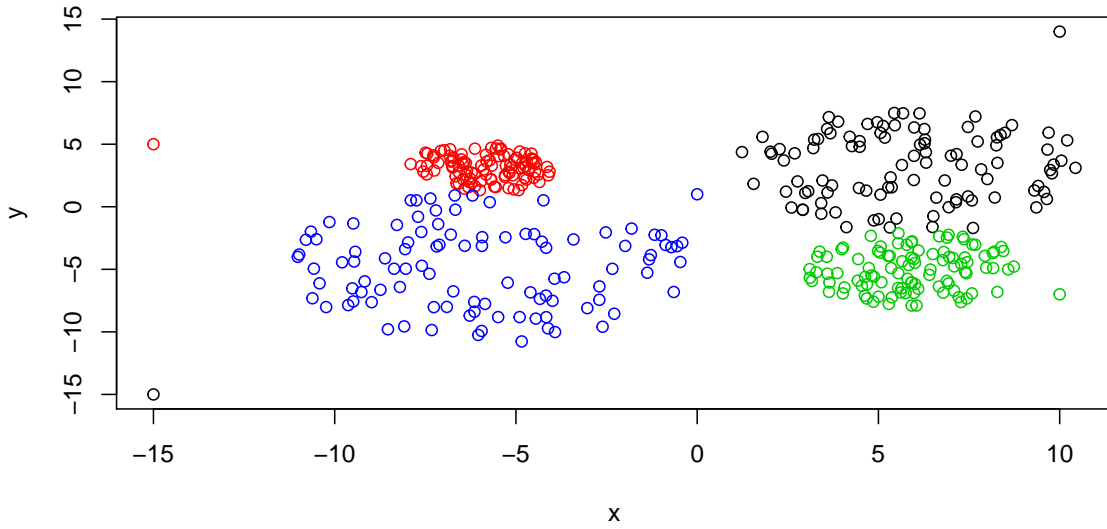
```
for(i in 1:4){  
  subs<- subset(data,subset = data[1] == i)  
  plot(subs[,c(3,4)],col =subs$class, xlab = "x",ylab = "y",main = paste("data set",i,sep= " "))  
}
```



data set 3



data set 4



2. Sea una matriz de disimilaridades o distancias $D_{n \times n}$ es una matriz tal que su elemento i, j es una disimilaridad $d(i, j)$ tal que $\forall i, j, k$:

- $d(i, j) \geq 0$
- $d(i, j) = 0$
- $d(i, j) = d(j, i)$
- $d(i, j) \leq d(i, k) + d(k, j)$

donde D es una matriz simetrica y su diagonal son 0.

Para la disimilaridad $d(i, j)$ representa una medida de la diferencia entre dos observaciones x_i y x_j en este caso usaremos la disimilaridad basada en distancia euclideana dado que no tenemos ninguna evidencia que la diferencia entre los individuos sea diferente de 0:

$$d(i, j) = \sqrt{\sum_{i=1}^p (x_{ic} - x_{cj})^2}$$

especificamente el criterio de vecino mas cercano expresado como :

$$d_{UV} = \min(d_{ij}) : i \in U, j \in V$$

ya que, los conglomerados formados por este data set no poseen formas estrictamente esfericas y por lo tanto se ajustarian mas (teoricamente) las comparaciones individuales de vecino mas cercano.

3. Para cada dataset se generaron los siguientes dendrogramas(uno por cada metodo):

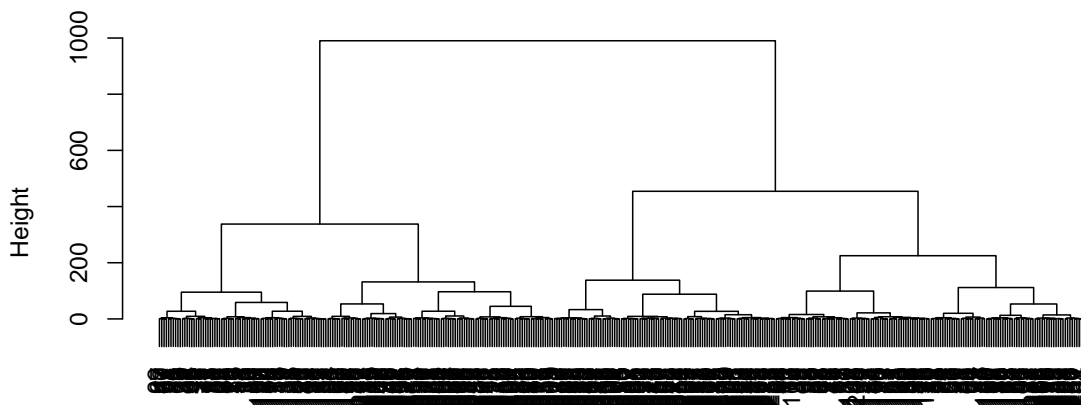
```
metodos <- c("ward.D", "ward.D2", "single", "complete", "average",
            "mcquitty", "median", "centroid")
for(i in metodos){
  subs<- subset(data,subset = data$id == 1)
  distanceE1 <- dist(subs[-c(1,2,5)],method = "euclidean")
  clusterE1 <- hclust(distanceE1,method = i)
  plot(clusterE1, main=paste("hclust para data set 1"))
  corteE1 <- cutree(clusterE1,k=length(unique(subs$class)))
  print(table(corteE1,subs$class))
  if(sum(diag(table(corteE1,subs$class))) == nrow(subs)){
    mejor1 <-clusterE1
    mejor1$corte <- corteE1
  }
  #-----
  subs<- subset(data,subset = data$id == 2)
  distanceE2 <- dist(subs[-c(1,2,5)],method = "euclidean")
  clusterE2 <- hclust(distanceE2,method = i)
  plot(clusterE2, main = paste("hclust para data set 2"))
  corteE2 <- cutree(clusterE2,k=length(unique(subs$class)))
  print(table(corteE2,subs$class))
  if(sum(diag(table(corteE2,subs$class))) == nrow(subs)){
    mejor2 <-clusterE2
    mejor2$corte <- corteE2
  }
  #-----
  subs<- subset(data,subset = data$id == 3)
  distanceE3 <- dist(subs[-c(1,2,5)],method = "euclidean")
  clusterE3 <- hclust(distanceE3,method = i)
  plot(clusterE3, main = paste("hclust para data set 3"))
  corteE3 <- cutree(clusterE3,k=length(unique(subs$class)))
  print(table(corteE3,subs$class))
  if(i=="ward.D"){
    mejor3 <- clusterE3
    mejor3$precision <- sum(diag(table(corteE3,subs$class)))
    mejor3$corte <- corteE3
  }
  if(sum(diag(table(corteE3,subs$class))) > mejor3$precision){
    mejor3 <- clusterE3
  }
}
```

```

    mejor3$precision <- sum(diag(table(corteE3,subs$class)))
    mejor3$corte <- corteE3
  }
  #-----
  subs<- subset(data,subset = data$id == 4)
  distanceE4 <- dist(subs[-c(1,2,5)],method = "euclidean")
  clusterE4 <- hclust(distanceE4,method = i)
  plot(clusterE4, main = paste("hclust para data set 4"))
  corteE4 <- cutree(clusterE4,k=length(unique(subs$class)))
  print(table(corteE4,subs$class))
  if(i=="ward.D"){
    mejor4 <- clusterE4
    mejor4$precision <- sum(diag(table(corteE4,subs$class)))
    mejor4$corte <- corteE4
  }
  if(sum(diag(table(corteE4,subs$class))) > mejor4$precision){
    mejor4 <- clusterE4
    mejor4$precision <- sum(diag(table(corteE4,subs$class)))
    mejor4$corte <- corteE4
  }
}

```

hclust para data set 1



distanceE1
hclust (*, "ward.D")

```

##
## corteE1  1  2  3  4
##          1 56 56 21  0
##          2 44 29  0  0
##          3  0 15 39 44
##          4  0  0 40 56

```

hclust para data set 2



distanceE2
hclust (*, "ward.D")

```
##
## corteE2   1   2   3   4
##         1 500   0   0   0
##         2   0 500   0   0
##         3   0   0 500   0
##         4   0   0   0 500
```

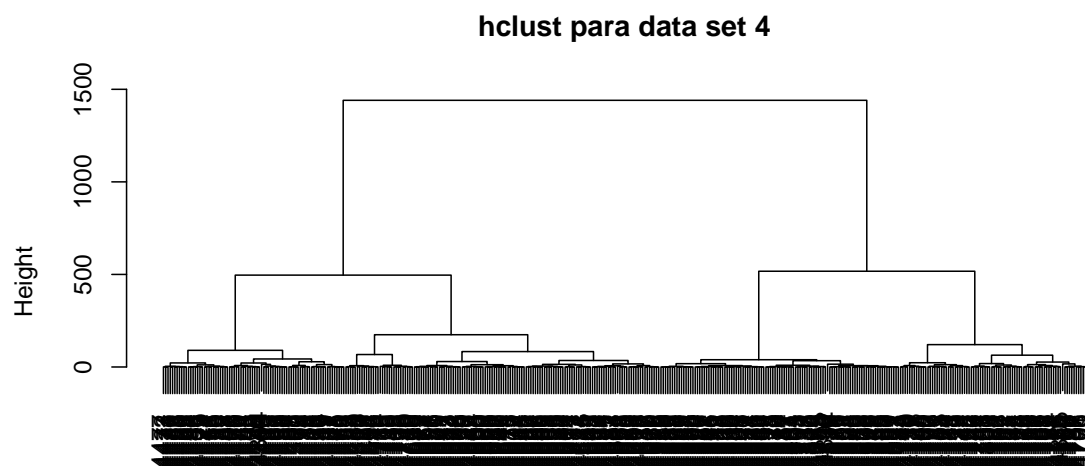
hclust para data set 3



distanceE3
hclust (*, "ward.D")

```
##
```

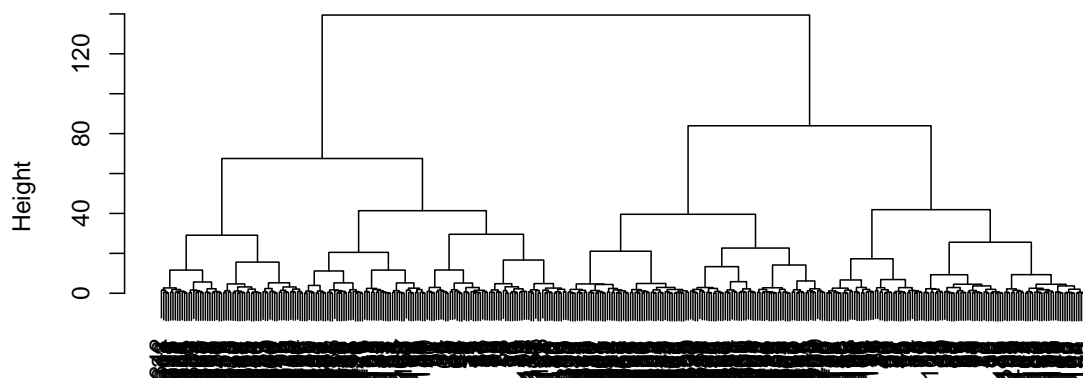
```
## corteE3  1  2  3  4
##          1 500  0 110  0
##          2  0 500  0  0
##          3  0  0 390  0
##          4  0  0  0 500
```



distanceE4
hclust (*, "ward.D")

```
##
## corteE4  1  2  3  4
##          1 80  0  0  0
##          2 21  0 101 16
##          3  0 101  0  4
##          4  1  0  0 81
```

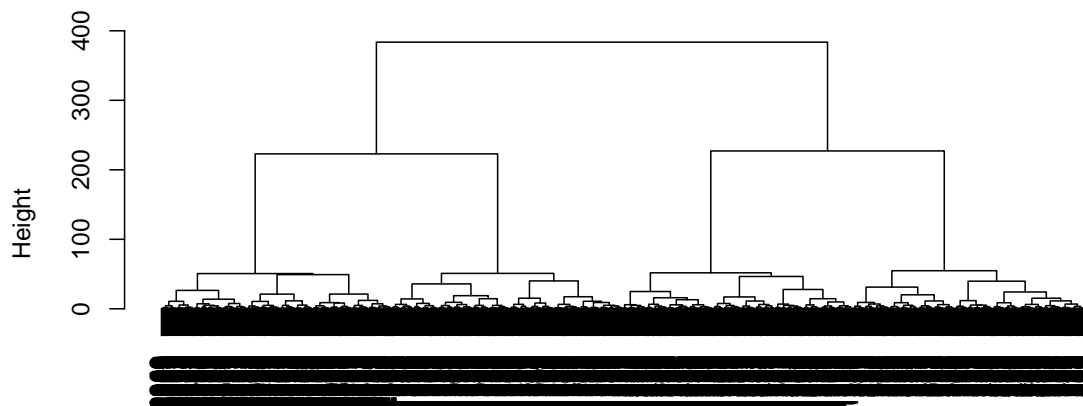
hclust para data set 1



distanceE1
hclust (*, "ward.D2")

```
##
## corteE1  1  2  3  4
##          1 56 56  0  0
##          2 44 44 26  0
##          3  0  0 56 56
##          4  0  0 18 44
```

hclust para data set 2



distanceE2
hclust (*, "ward.D2")

```
##
```



```
## corteE2   1   2   3   4
##          1 500   0   0   0
##          2   0 500   0   0
##          3   0   0 500   0
##          4   0   0   0 500
```

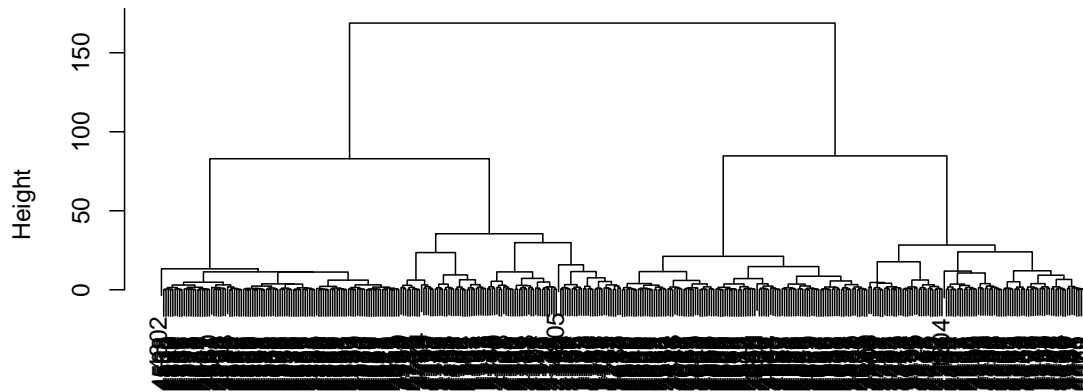
hclust para data set 3



```
distanceE3
hclust (*, "ward.D2")
```

```
##
## corteE3   1   2   3   4
##          1 95   0 500   0
##          2 405   0   0   0
##          3   0 500   0   0
##          4   0   0   0 500
```

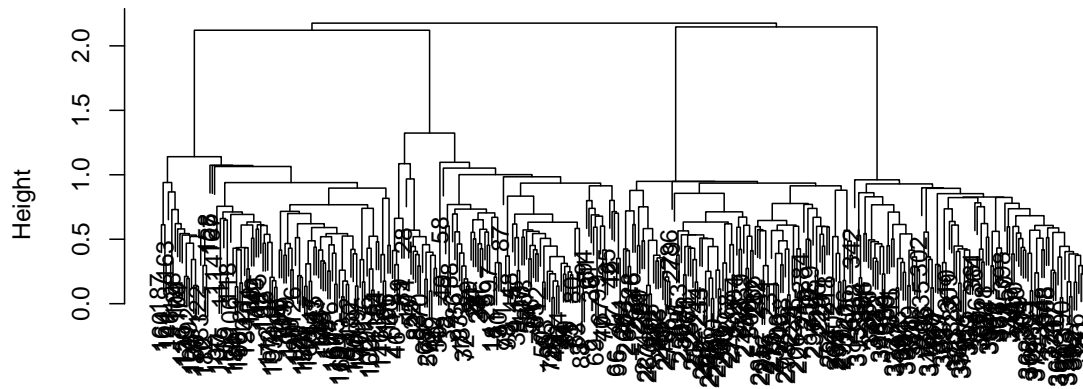
hclust para data set 4



distanceE4
hclust (*, "ward.D2")

```
##
## corteE4    1    2    3    4
##          1  94    0    0    1
##          2   7    0 101    0
##          3   0 101    0    4
##          4   1    0    0  96
```

hclust para data set 1

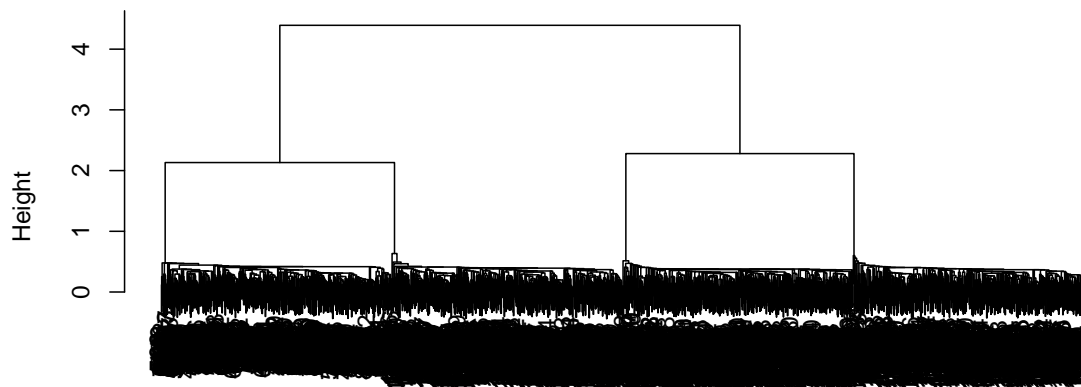


distanceE1
hclust (*, "single")

```
##
```

```
## corteE1  1  2  3  4
##          1 100  0  0  0
##          2  0 100  0  0
##          3  0  0 100  0
##          4  0  0  0 100
```

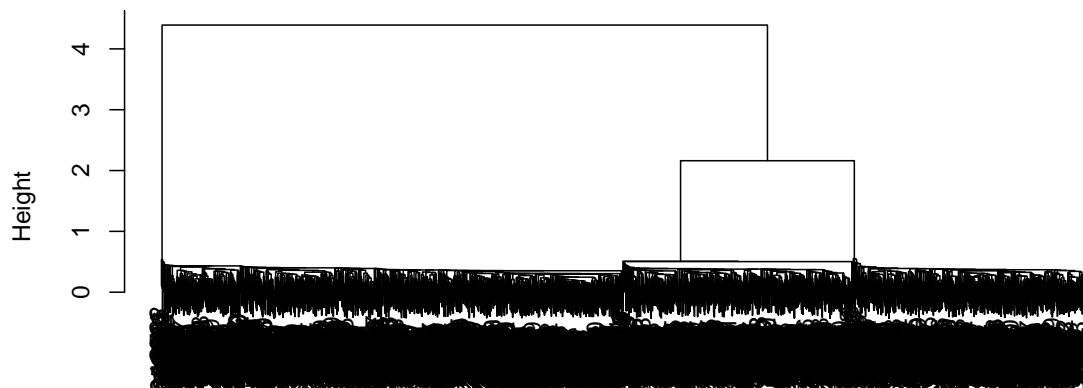
hclust para data set 2



distanceE2
hclust (*, "single")

```
##
## corteE2  1  2  3  4
##          1 500  0  0  0
##          2  0 500  0  0
##          3  0  0 500  0
##          4  0  0  0 500
```

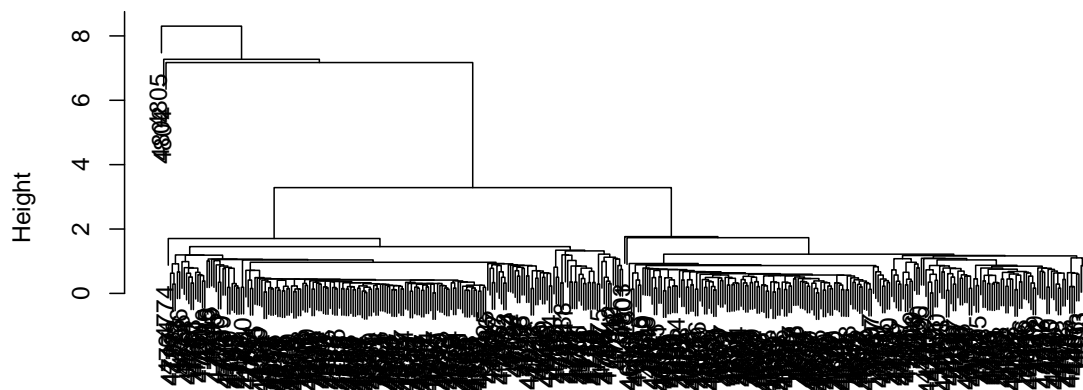
hclust para data set 3



distanceE3
hclust (*, "single")

```
##
## corteE3    1    2    3    4
##      1 500    0 500    0
##      2    0 499    0    0
##      3    0    1    0    0
##      4    0    0    0 500
```

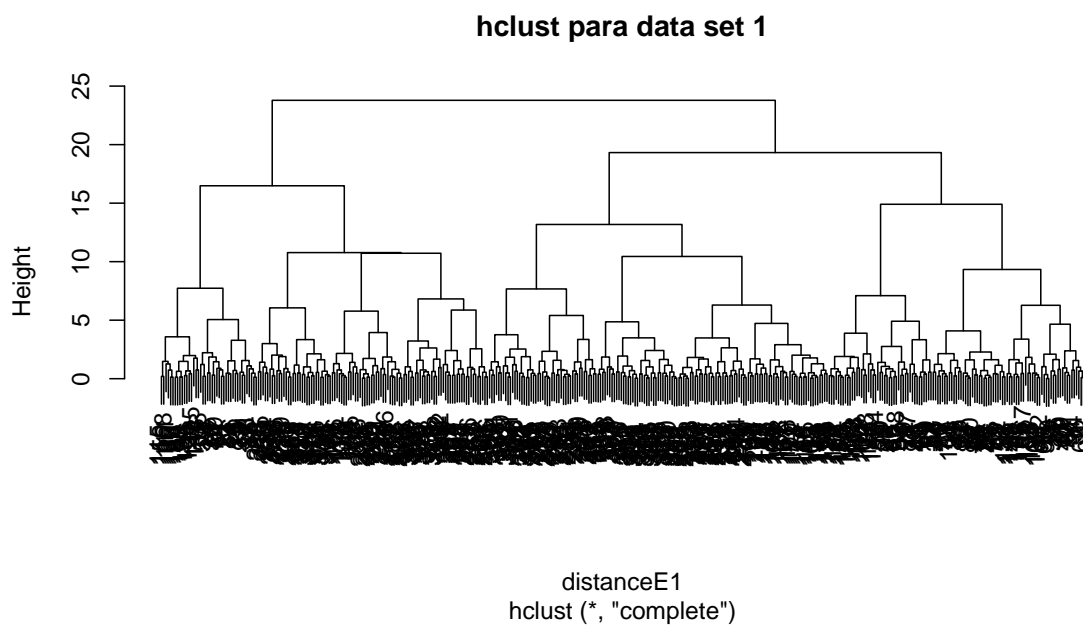
hclust para data set 4



distanceE4
hclust (*, "single")

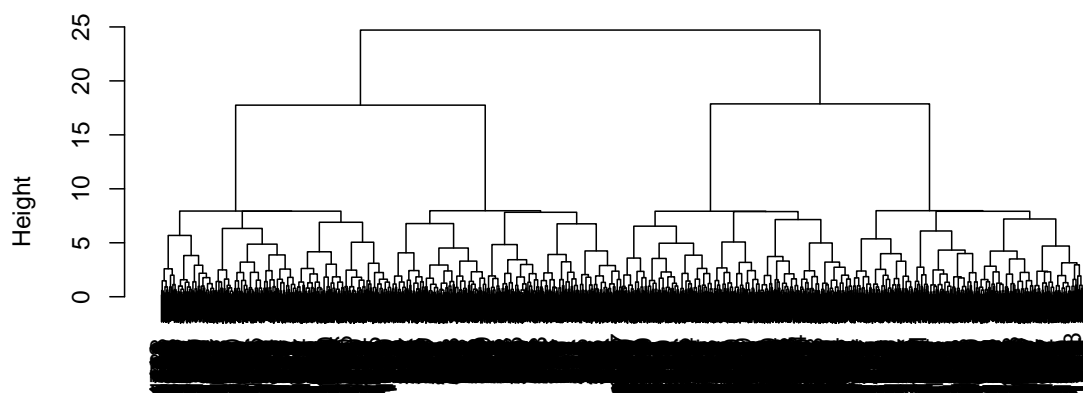
```
##
```

```
## corteE4  1  2  3  4
##          1 100 100 101 101
##          2  0  1  0  0
##          3  1  0  0  0
##          4  1  0  0  0
```



```
##
## corteE1  1  2  3  4
##          1 75 36  0  0
##          2 25 17  0  0
##          3  0 35 56 56
##          4  0 12 44 44
```

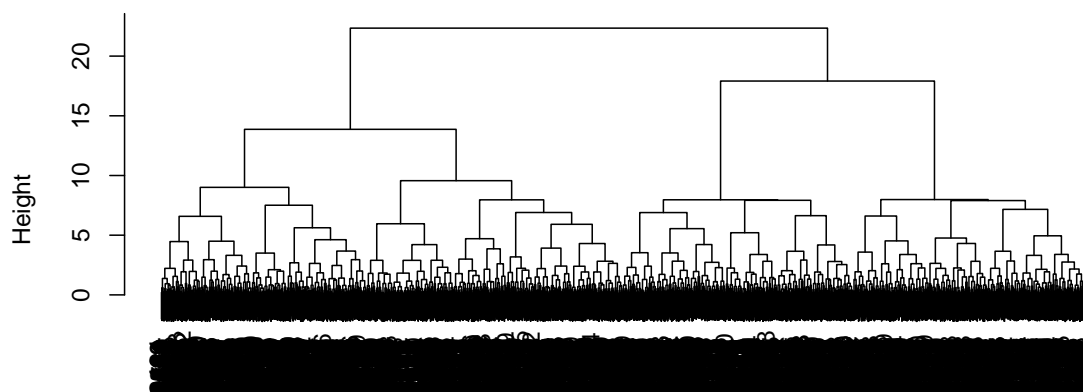
hclust para data set 2



distanceE2
hclust (*, "complete")

```
##
## corteE2   1   2   3   4
##          1 500   0   0   0
##          2   0 500   0   0
##          3   0   0 500   0
##          4   0   0   0 500
```

hclust para data set 3

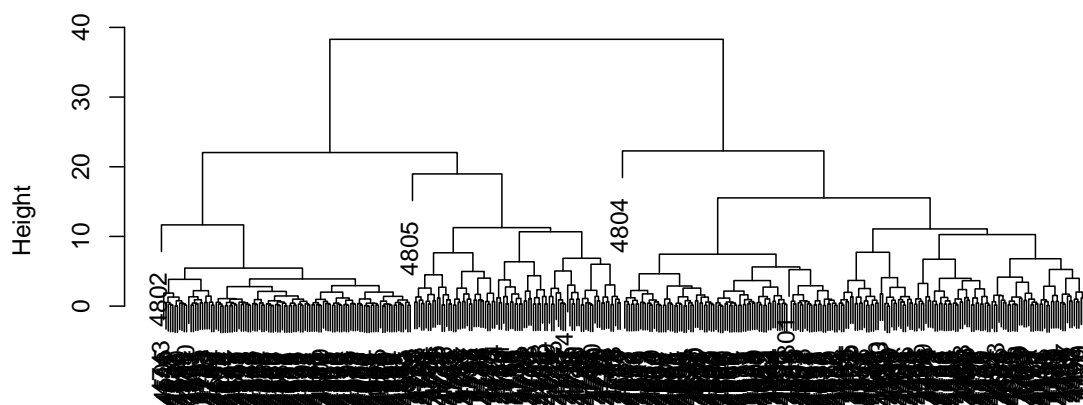


distanceE3
hclust (*, "complete")

```
##
```

```
## corteE3   1   2   3   4
##          1 470   0  85   0
##          2  30   0 415   0
##          3   0 500   0   0
##          4   0   0   0 500
```

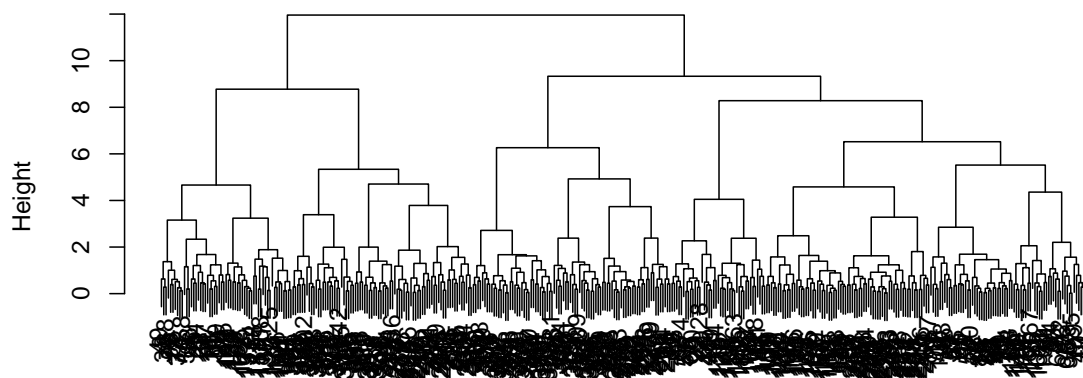
hclust para data set 4



distanceE4
hclust (*, "complete")

```
##
## corteE4   1   2   3   4
##          1 100   0 101   1
##          2   0 101   0   9
##          3   1   0   0  91
##          4   1   0   0   0
```

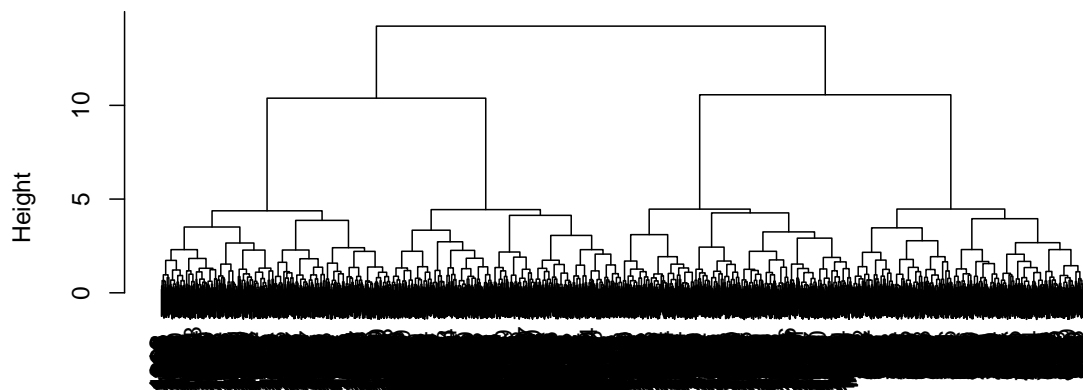
hclust para data set 1



distanceE1
hclust (*, "average")

```
##
## corteE1  1  2  3  4
##          1 72 71 36  0
##          2 28 29  0  0
##          3  0  0 30 56
##          4  0  0 34 44
```

hclust para data set 2



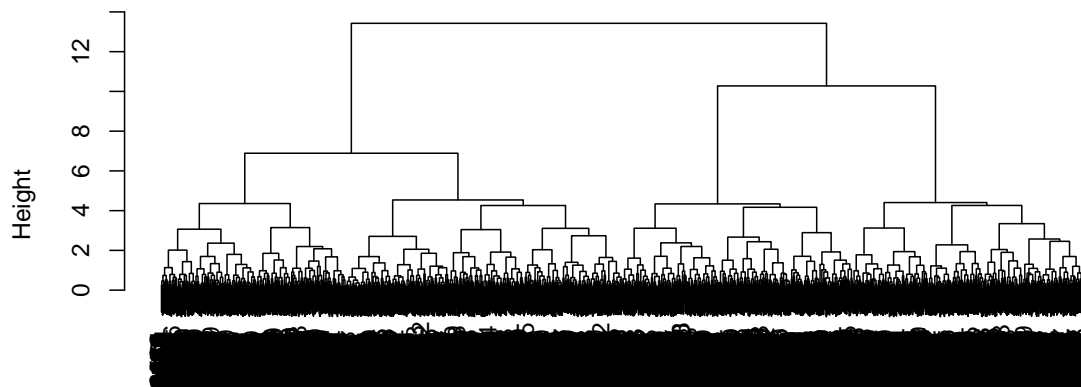
distanceE2
hclust (*, "average")

```
##
```



```
## corteE2    1    2    3    4
##           1 500    0    0    0
##           2    0 500    0    0
##           3    0    0 500    0
##           4    0    0    0 500
```

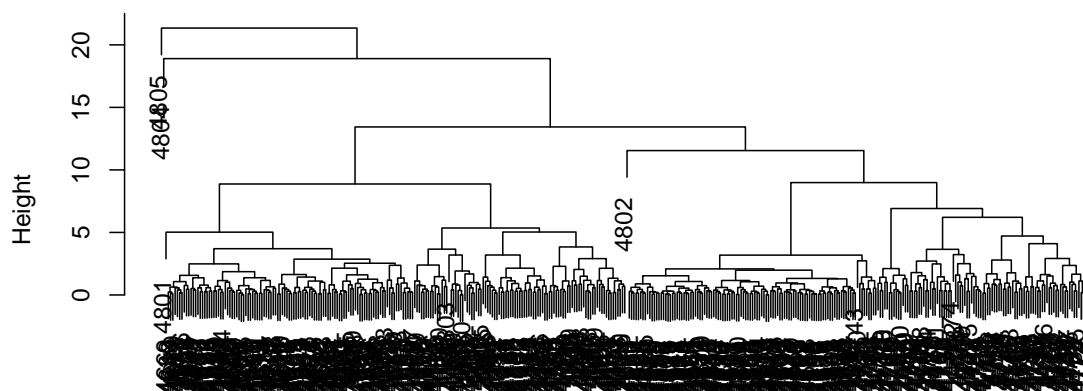
hclust para data set 3



distanceE3
hclust (*, "average")

```
##
## corteE3    1    2    3    4
##           1 500    0 99    0
##           2    0 500    0    0
##           3    0    0 401    0
##           4    0    0    0 500
```

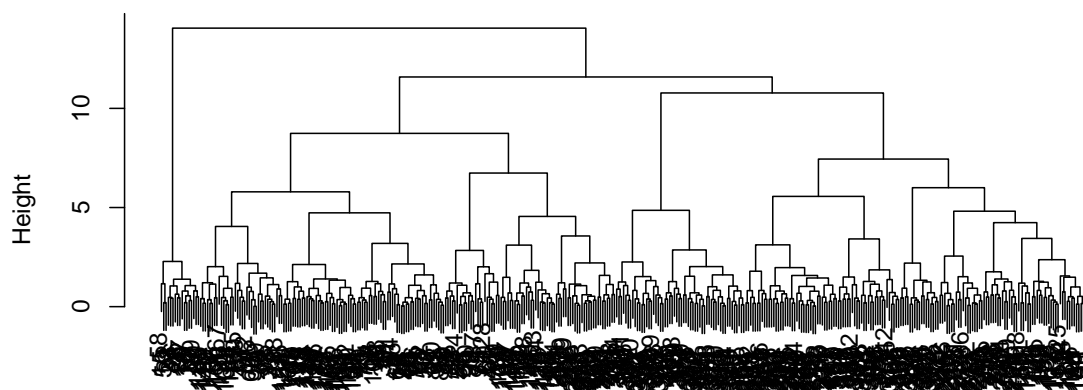
hclust para data set 4



distanceE4
hclust (*, "average")

```
##
## corteE4   1   2   3   4
##          1 100   0 101   1
##          2   0 101   0 100
##          3   1   0   0   0
##          4   1   0   0   0
```

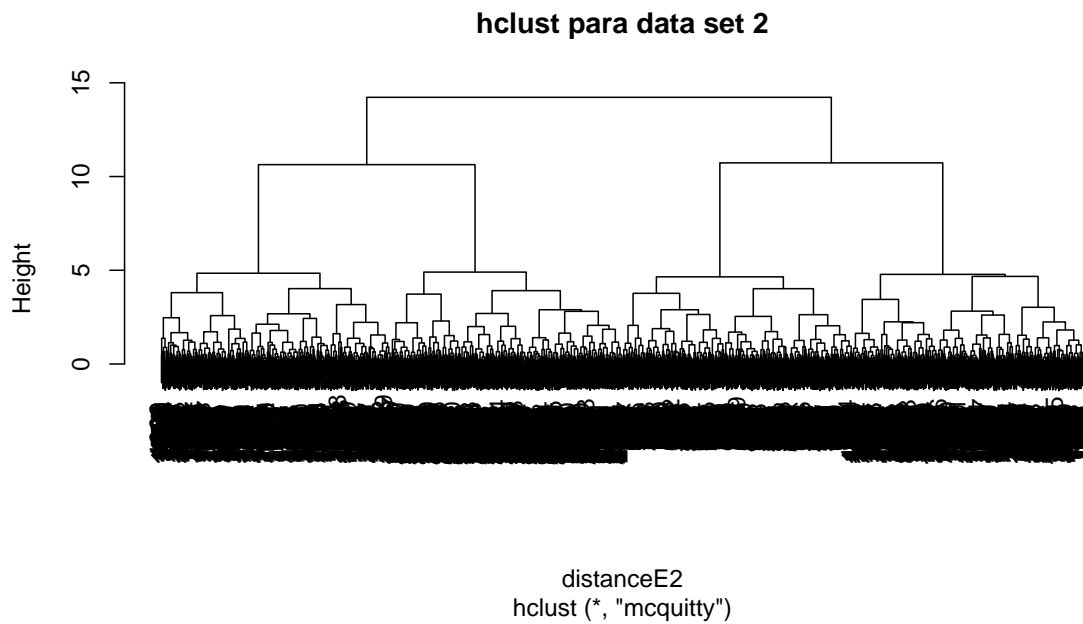
hclust para data set 1



distanceE1
hclust (*, "mcquitty")

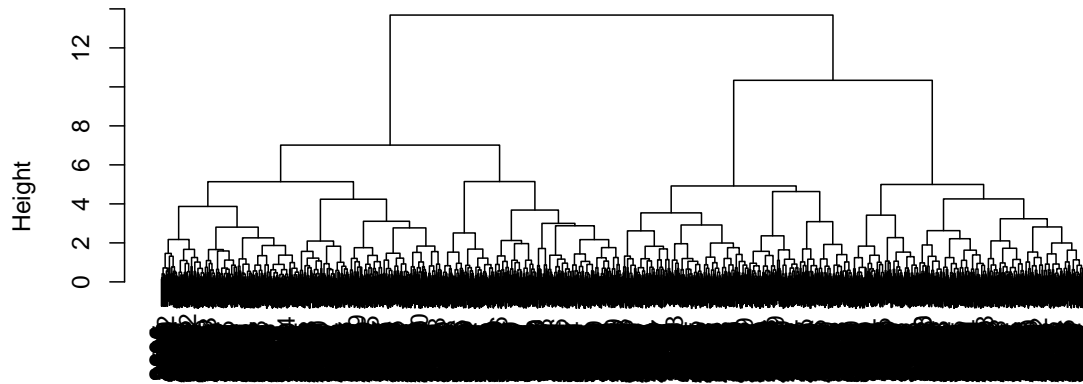
```
##
```

```
## corteE1  1  2  3  4
##          1 83 71 26  0
##          2 17  0  0  0
##          3  0 29 74 44
##          4  0  0  0 56
```



```
##
## corteE2   1  2  3  4
##           1 500  0  0  0
##           2  0 500  0  0
##           3  0  0 500  0
##           4  0  0  0 500
```

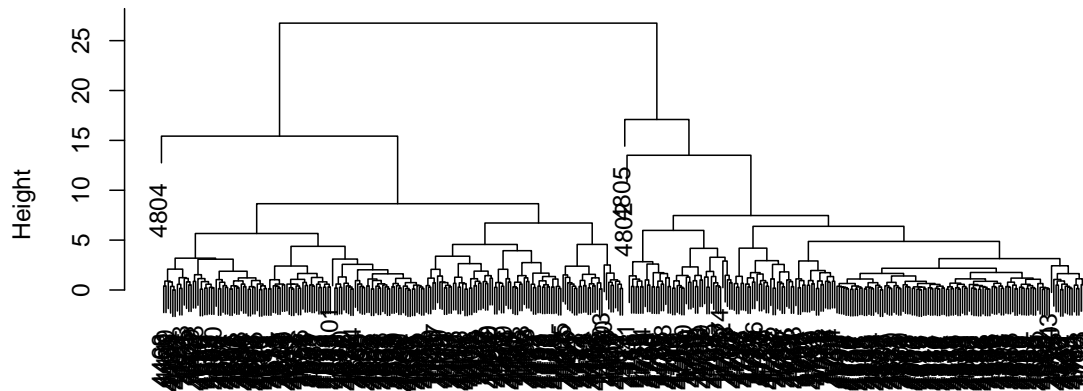
hclust para data set 3



distanceE3
hclust (*, "mcquitty")

```
##
## corteE3   1   2   3   4
##          1 498   0 112   0
##          2   2   0 388   0
##          3   0 500   0   0
##          4   0   0   0 500
```

hclust para data set 4

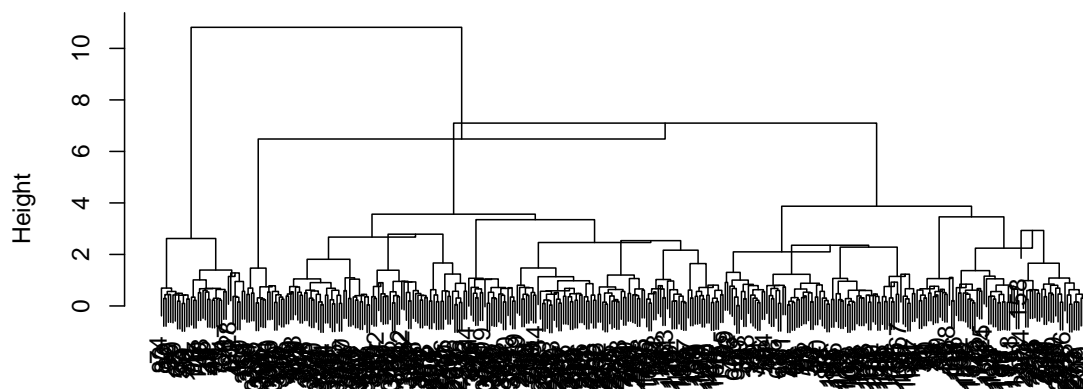


distanceE4
hclust (*, "mcquitty")

```
##
```

```
## corteE4  1  2  3  4
##          1 100  0 101  1
##          2  0 101  0 100
##          3  1  0  0  0
##          4  1  0  0  0
```

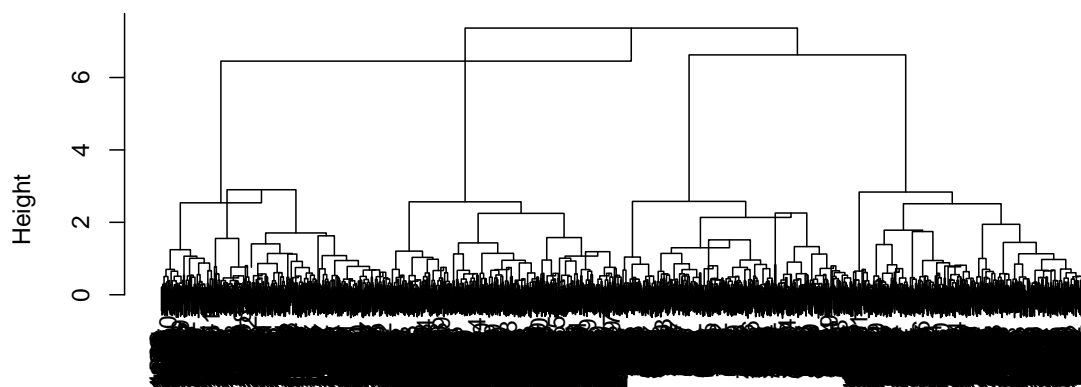
hclust para data set 1



distanceE1
hclust (*, "median")

```
##
## corteE1  1  2  3  4
##          1 37  0  0  0
##          2 63 66 27  0
##          3  0 34 73 83
##          4  0  0  0 17
```

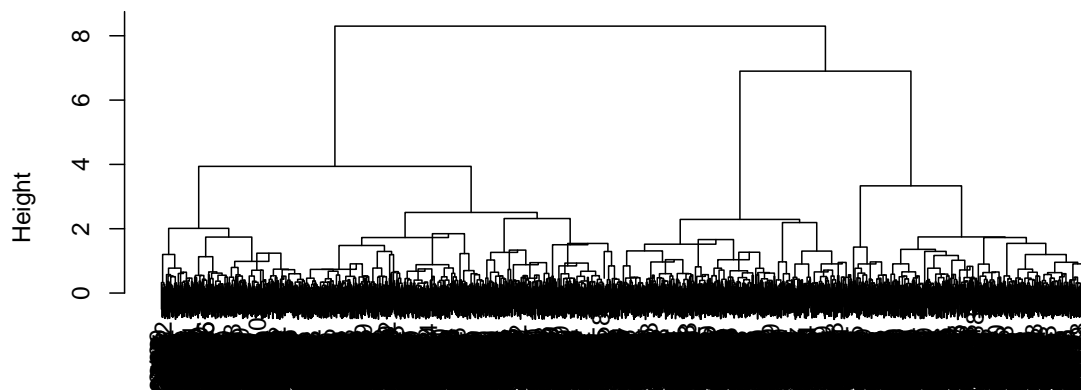
hclust para data set 2



distanceE2
hclust (*, "median")

```
##
## corteE2    1    2    3    4
##          1 500    0    0    0
##          2    0 500    0    0
##          3    0    0 500    0
##          4    0    0    0 500
```

hclust para data set 3

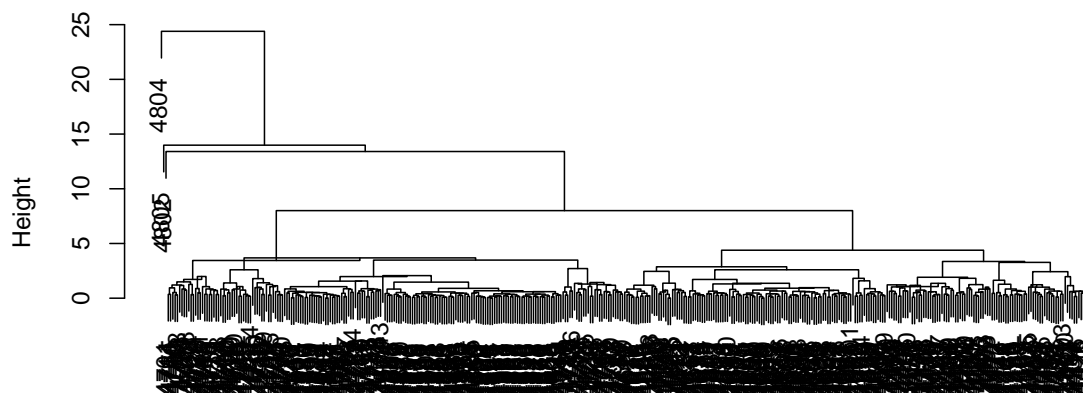


distanceE3
hclust (*, "median")

```
##
```

```
## corteE3   1   2   3   4
##          1 188   0 500   0
##          2 312   0   0   0
##          3   0 500   0   0
##          4   0   0   0 500
```

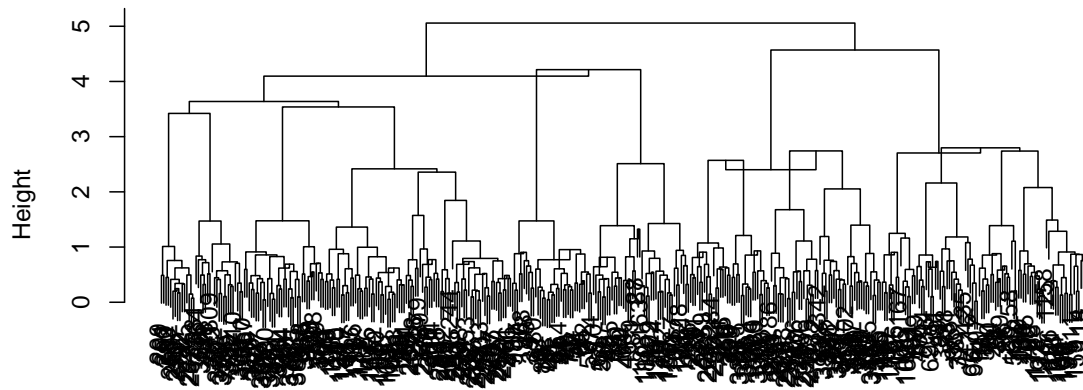
hclust para data set 4



```
distanceE4
hclust (*, "median")
```

```
##
## corteE4   1   2   3   4
##          1 100 100 101 101
##          2   0   1   0   0
##          3   1   0   0   0
##          4   1   0   0   0
```

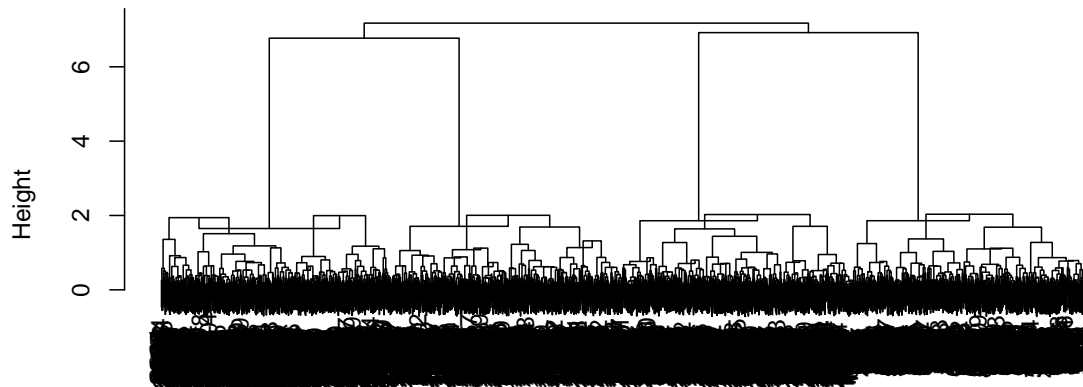
hclust para data set 1



distanceE1
hclust (*, "centroid")

```
##
## corteE1  1  2  3  4
##          1 56 21  0  0
##          2 44 44  0  0
##          3  0 35 61 56
##          4  0  0 39 44
```

hclust para data set 2



distanceE2
hclust (*, "centroid")

```
##
```



```
## corteE2  1  2  3  4
##          1 500  0  0  0
##          2  0 500  0  0
##          3  0  0 500  0
##          4  0  0  0 500
```

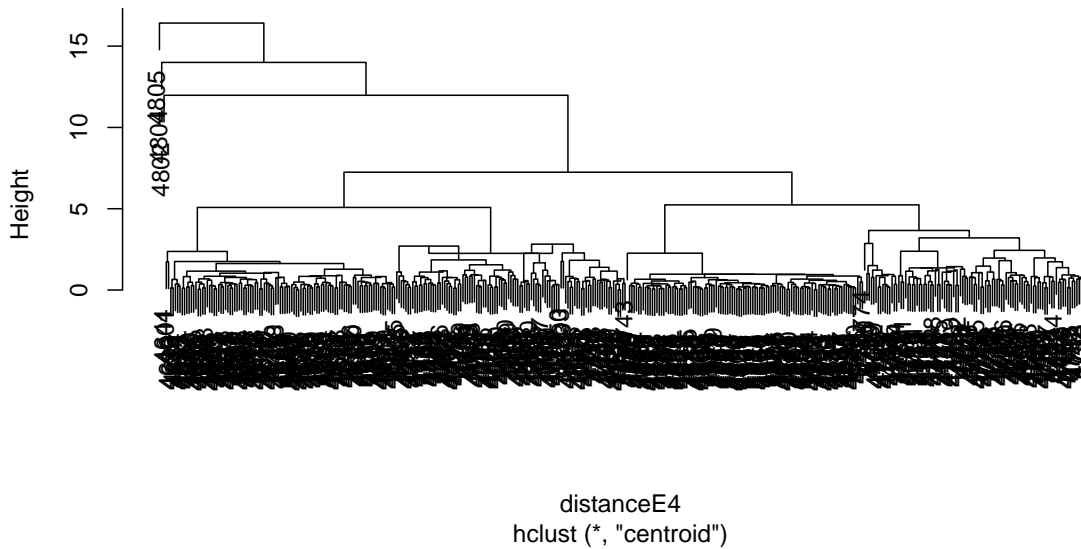
hclust para data set 3



distanceE3
hclust (*, "centroid")

```
##
## corteE3  1  2  3  4
##          1 500  0 133  0
##          2  0 500  0  0
##          3  0  0 367  0
##          4  0  0  0 500
```

hclust para data set 4



```
##
## corteE4    1    2    3    4
##           1 100 100 101 101
##           2   0   1   0   0
##           3   1   0   0   0
##           4   1   0   0   0
```

Para decidir el mejor dendrograma se selecciono encontrando aquel que maximizara la suma de las diagonales, i.e., la precisión al clasificar cada clase como realmente esta clasificada en el set de datos original.

```
print("Para el set de datos 1 el mejor dendrograma fue el que poseyo las siguientes configuraciones")
```

```
## [1] "Para el set de datos 1 el mejor dendrograma fue el que poseyo las siguientes configuraciones"
```

```
mejor1
```

```
##
## Call:
## hclust(d = distanceE1, method = i)
##
## Cluster method   : single
## Distance         : euclidean
## Number of objects: 400
```

```
print("Para el set de datos 2 el mejor dendrograma fue el que poseyo las siguientes configuraciones")
```

```
## [1] "Para el set de datos 2 el mejor dendrograma fue el que poseyo las siguientes configuraciones"
```

```
mejor2
```

```
##  
## Call:  
## hclust(d = distanceE2, method = i)  
##  
## Cluster method   : centroid  
## Distance         : euclidean  
## Number of objects: 2000
```

```
print("Para el set de datos 3 el mejor dendrograma fue el que poseyo las siguientes configuraciones")
```

```
## [1] "Para el set de datos 3 el mejor dendrograma fue el que poseyo las siguientes configuraciones"
```

```
mejor3
```

```
##  
## Call:  
## hclust(d = distanceE3, method = i)  
##  
## Cluster method   : average  
## Distance         : euclidean  
## Number of objects: 2000
```

```
print("Para el set de datos 4 el mejor dendrograma fue el que poseyo las siguientes configuraciones")
```

```
## [1] "Para el set de datos 4 el mejor dendrograma fue el que poseyo las siguientes configuraciones"
```

```
mejor4
```

```
##  
## Call:  
## hclust(d = distanceE4, method = i)  
##  
## Cluster method   : complete  
## Distance         : euclidean  
## Number of objects: 405
```

4. Sea cada dendrograma ganador

```
for(i in 2:5){  
  print("para el set de datos 1 el dendrograma debe tener altura")  
  print(sort(mejor1$height,decreasing = T)[i])  
  print(paste("para i =",i," cluster"))  
  print("para el set de datos 2 el dendrograma debe tener altura")  
  print(sort(mejor2$height,decreasing = T)[i])  
  print(paste("para i =",i," cluster"))  
}
```

```

print("para el set de datos 3 el dendrograma debe tener altura")
print(sort(mejor3$height,decreasing = T)[i])
print(paste("para i =",i," cluster"))
print("para el set de datos 4 el dendrograma debe tener altura")
print(sort(mejor4$height,decreasing = T)[i])
print(paste("para i =",i," cluster"))
}

```

```

## [1] "para el set de datos 1 el dendrograma debe tener altura"
## [1] 2.147032
## [1] "para i = 2 cluster"
## [1] "para el set de datos 2 el dendrograma debe tener altura"
## [1] 6.919186
## [1] "para i = 2 cluster"
## [1] "para el set de datos 3 el dendrograma debe tener altura"
## [1] 10.27625
## [1] "para i = 2 cluster"
## [1] "para el set de datos 4 el dendrograma debe tener altura"
## [1] 22.27385
## [1] "para i = 2 cluster"
## [1] "para el set de datos 1 el dendrograma debe tener altura"
## [1] 2.121351
## [1] "para i = 3 cluster"
## [1] "para el set de datos 2 el dendrograma debe tener altura"
## [1] 6.769959
## [1] "para i = 3 cluster"
## [1] "para el set de datos 3 el dendrograma debe tener altura"
## [1] 6.888918
## [1] "para i = 3 cluster"
## [1] "para el set de datos 4 el dendrograma debe tener altura"
## [1] 22.04079
## [1] "para i = 3 cluster"
## [1] "para el set de datos 1 el dendrograma debe tener altura"
## [1] 1.323578
## [1] "para i = 4 cluster"
## [1] "para el set de datos 2 el dendrograma debe tener altura"
## [1] 2.033709
## [1] "para i = 4 cluster"
## [1] "para el set de datos 3 el dendrograma debe tener altura"
## [1] 4.537209
## [1] "para i = 4 cluster"
## [1] "para el set de datos 4 el dendrograma debe tener altura"
## [1] 18.97112
## [1] "para i = 4 cluster"
## [1] "para el set de datos 1 el dendrograma debe tener altura"
## [1] 1.140298
## [1] "para i = 5 cluster"
## [1] "para el set de datos 2 el dendrograma debe tener altura"
## [1] 2.02526
## [1] "para i = 5 cluster"
## [1] "para el set de datos 3 el dendrograma debe tener altura"

```

```
## [1] 4.406336
## [1] "para i = 5 cluster"
## [1] "para el set de datos 4 el dendrograma debe tener altura"
## [1] 15.53133
## [1] "para i = 5 cluster"
```

5. dado que las clases de los sets de datos de entrada poseen todos 4 clases, luego seleccionemos este valor como k para encontrar un arbol en cuya altura produzca k numero de clúster

```
sort(mejor1$height,decreasing = T)[4]
```

```
## [1] 1.323578
```

```
sort(mejor2$height,decreasing = T)[4]
```

```
## [1] 2.033709
```

```
sort(mejor3$height,decreasing = T)[4]
```

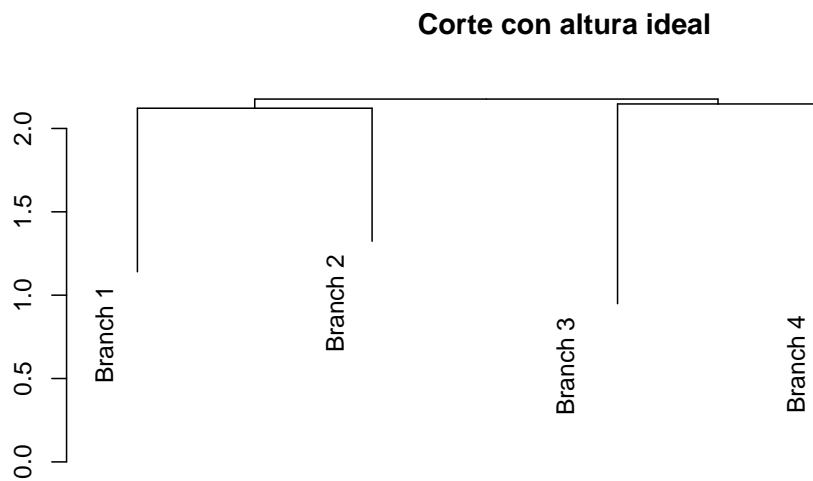
```
## [1] 4.537209
```

```
sort(mejor4$height,decreasing = T)[4]
```

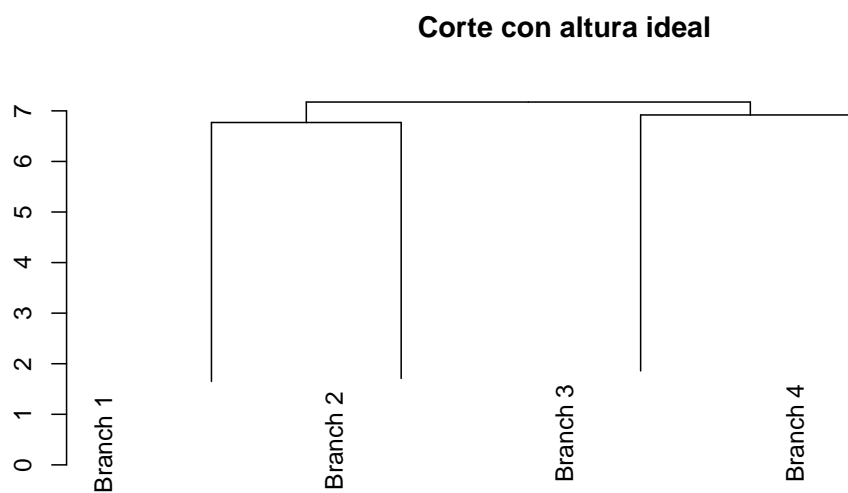
```
## [1] 18.97112
```

6. Grafica de los dendrogramas ganadores segun el mejor numero de altura (cada corte representa la parte superior del arbol cortado en dicha altura):

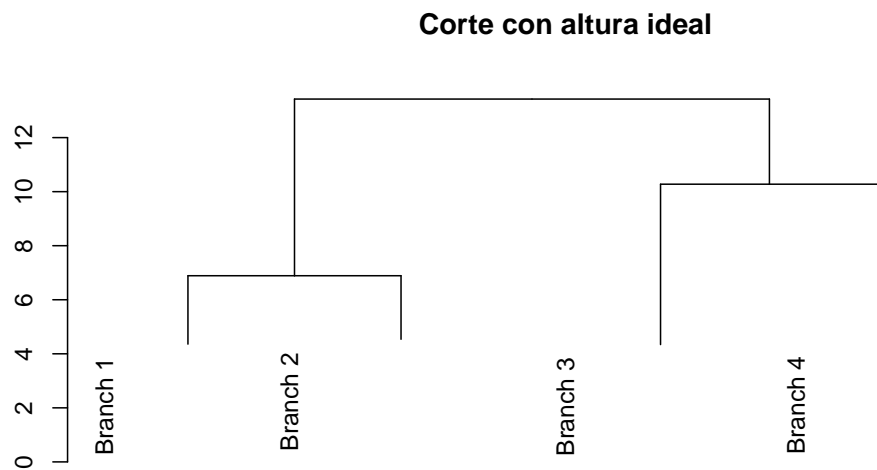
```
mejor1$dendo <- as.dendrogram(mejor1)
mejorcorte1 <- cut(mejor1$dendo,h= sort(mejor1$height,decreasing = T)[4])$upper
plot(mejorcorte1, main = "Corte con altura ideal")
```



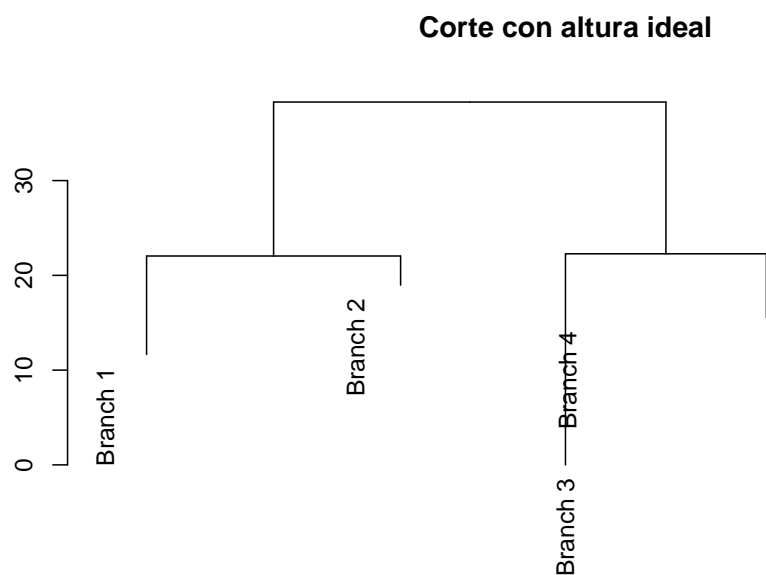
```
mejor2$dendo <- as.dendrogram(mejor2)
mejorcorte2 <- cut(mejor2$dendo, h= sort(mejor2$height, decreasing = T)[4])$upper
plot(mejorcorte2, main = "Corte con altura ideal")
```



```
mejor3$dendo <- as.dendrogram(mejor3)
mejorcorte3 <- cut(mejor3$dendo, h= sort(mejor3$height, decreasing = T)[4])$upper
plot(mejorcorte3, main = "Corte con altura ideal")
```



```
mejor4$dendo <- as.dendrogram(mejor4)
mejorcorte4 <- cut(mejor4$dendo, h= sort(mejor4$height, decreasing = T)[4])$upper
plot(mejorcorte4, main = "Corte con altura ideal")
```



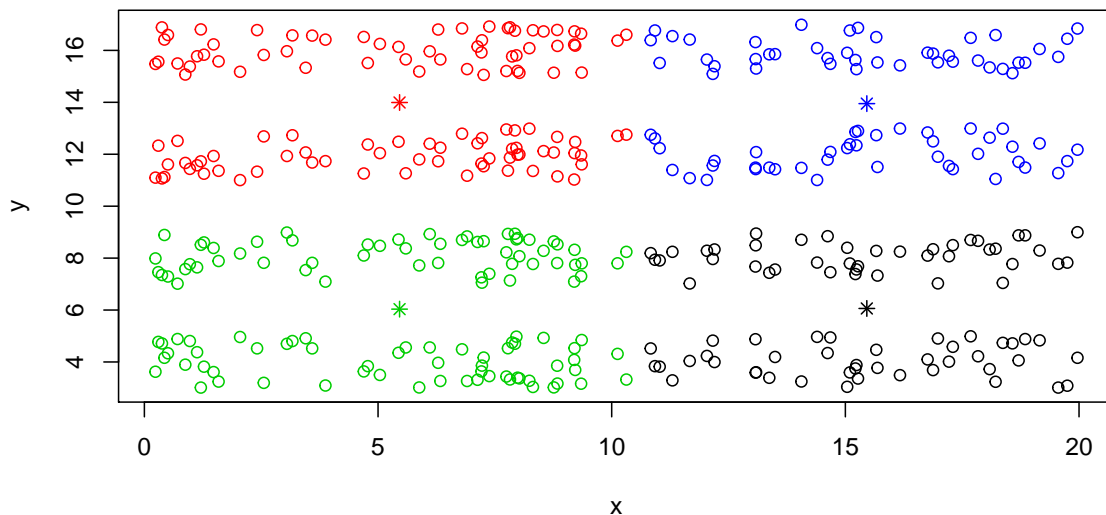
7. Dado que se conoce la clasificación en 4 diferentes conjuntos de la data se usara el $k=4$
8. Grafica de la clusterizacion mediante k-medias y sus centros:

```

for(i in 1:4){
  subs<- subset(data,subset = data$id == i)
  kmd <- kmeans(subs[c(3,4)],4)
  plot(subs[c(3,4)], col = kmd$cluster, xlab = "x",ylab = "y",
        main = paste("Set de datos",i,sep=" "))
  points(kmd$centers, col = 1:4, pch = 8)
  print("matriz de confusion de aplicacion de k-medias")
  print(table(kmd$cluster,subs$class))
  if(i==1) {
    print("matriz de confusion de aplicacion de clasificacion por aglomeración")
    print(table(mejor1$corte,subs$class))}
  if(i==2) {
    print("matriz de confusion de aplicacion de clasificacion por aglomeración")
    print(table(mejor2$corte,subs$class))}
  if(i==3) {
    print("matriz de confusion de aplicacion de clasificacion por aglomeración")
    print(table(mejor3$corte,subs$class))}
  if(i==4) {
    print("matriz de confusion de aplicacion de clasificacion por aglomeración")
    print(table(mejor4$corte,subs$class))}
}

```

Set de datos 1



```

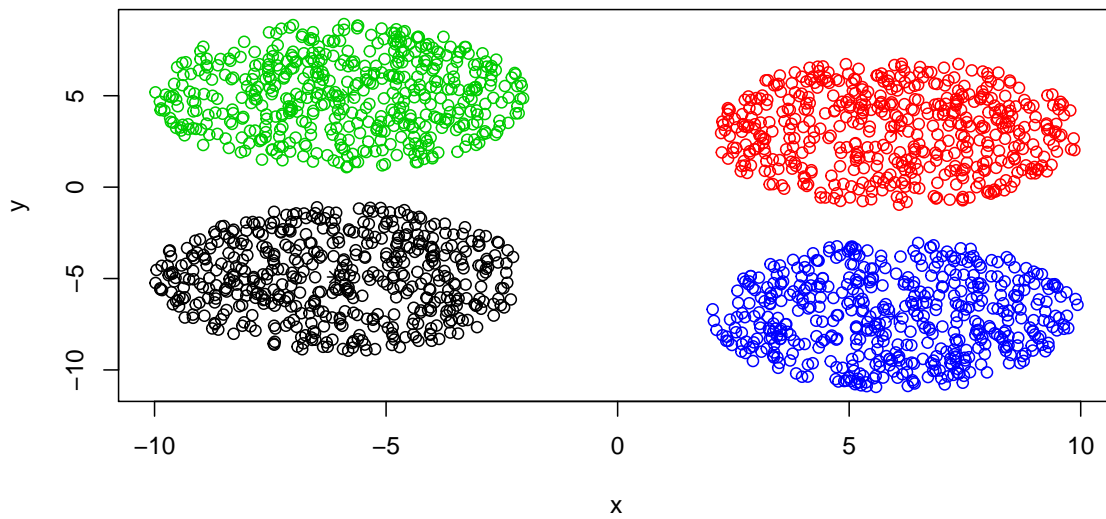
## [1] "matriz de confusion de aplicacion de k-medias"
##
##      1  2  3  4
## 1 42 42  0  0
## 2  0  0 58 58
## 3 58 58  0  0
## 4  0  0 42 42

```



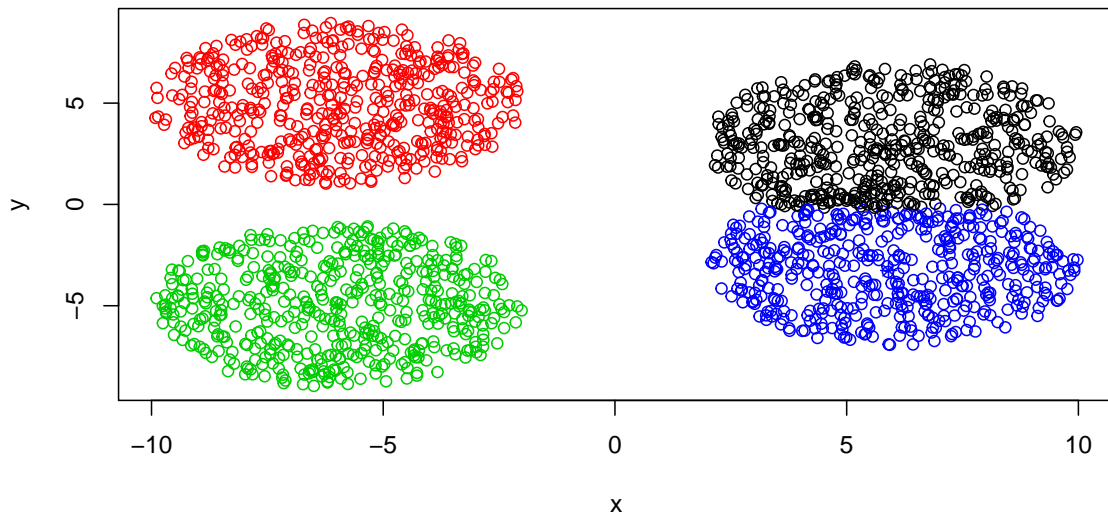
```
## [1] "matriz de confusion de aplicacion de clasificacion por aglomeración"
##
##      1    2    3    4
## 1 100    0    0    0
## 2    0 100    0    0
## 3    0    0 100    0
## 4    0    0    0 100
```

Set de datos 2

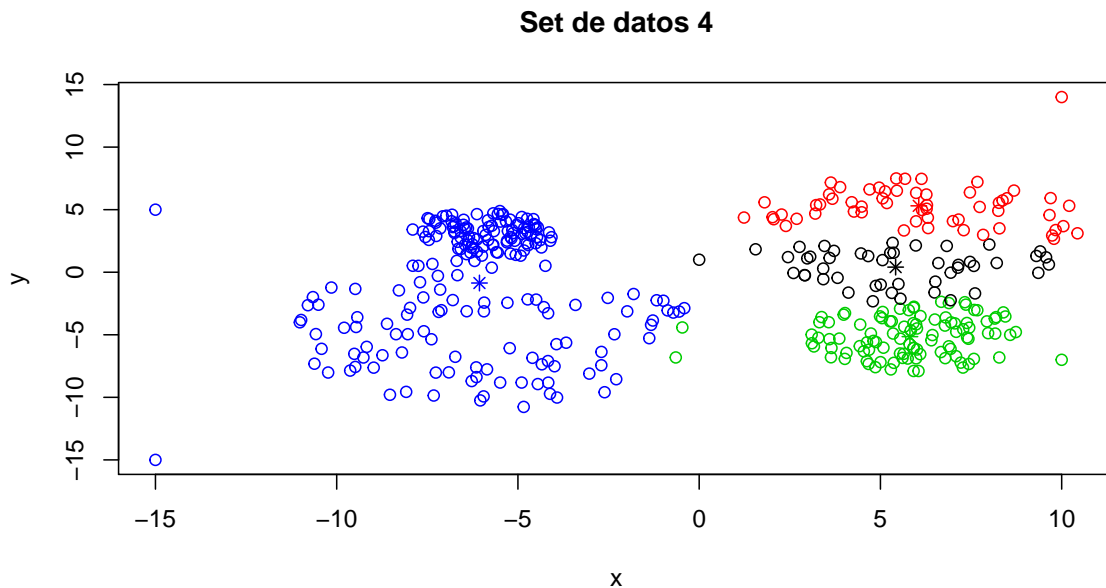


```
## [1] "matriz de confusion de aplicacion de k-medias"
##
##      1    2    3    4
## 1    0    0    0 500
## 2 500    0    0    0
## 3    0 500    0    0
## 4    0    0 500    0
## [1] "matriz de confusion de aplicacion de clasificacion por aglomeración"
##
##      1    2    3    4
## 1 500    0    0    0
## 2    0 500    0    0
## 3    0    0 500    0
## 4    0    0    0 500
```

Set de datos 3



```
## [1] "matriz de confusion de aplicacion de k-medias"
##
##      1   2   3   4
## 1 478   0  47   0
## 2   0 500   0   0
## 3   0   0   0 500
## 4  22   0 453   0
## [1] "matriz de confusion de aplicacion de clasificacion por aglomeración"
##
##      1   2   3   4
## 1 500   0  99   0
## 2   0 500   0   0
## 3   0   0 401   0
## 4   0   0   0 500
```



```
## [1] "matriz de confusion de aplicacion de k-medias"
##
##      1   2   3   4
## 1  43   0   3   1
## 2  58   0   0   0
## 3   0   0  98   2
## 4   1 101   0  98
## [1] "matriz de confusion de aplicacion de clasificacion por aglomeración"
##
##      1   2   3   4
## 1 100   0 101   1
## 2   0 101   0   9
## 3   1   0   0  91
## 4   1   0   0   0
```

9. Para set de datos en los cuales la ubicación de los puntos o individuos posee una forma esferica puede ser conveniente el uso de k-medias, y para clusterizacion de formas no-esfericas proveen mayor precision los metodos de clasificación jerárquica por aglomeración.

Por ejemplo para el set 4 genera diferentes combinaciones de cluster y de centros.

10. para calcular la clasificacion de la nueva instancia la agregamos a los sets y corremos los algoritmos de clusterizacion respectivos.

```
subs<- subset(data,subset = data$id == 1)
subs1 <- rbind(subs[, -c(1,2,5)], c(3,3))
distanceE1 <- dist(subs1,method = "euclidean")
clusterE1 <- hclust(distanceE1,method = mejor1$method)
corteE1 <- cutree(clusterE1,k=length(unique(subs$class)))
print("en el Set 1")
```

```
## [1] "en el Set 1"
```

```
print(paste("en clasificacion jerarquica por aglomeracion",corteE1[length(corteE1)]))
```

```
## [1] "en clasificacion jerarquica por aglomeracion 1"
```

```
kmd <- kmeans(subs1,4)
print(paste("en K-medias por otro lado reultaria en el cluster",unique(kmd$cluster)))
```

```
## [1] "en K-medias por otro lado reultaria en el cluster 2"
## [2] "en K-medias por otro lado reultaria en el cluster 4"
## [3] "en K-medias por otro lado reultaria en el cluster 1"
## [4] "en K-medias por otro lado reultaria en el cluster 3"
```

```
#-----
subs<- subset(data,subset = data$id == 2)
subs2 <- rbind(subs[, -c(1,2,5)],c(3,3))
distanceE2 <- dist(subs2,method = "euclidean")
clusterE2 <- hclust(distanceE2,method = mejor2$method)
corteE2 <- cutree(clusterE2,k=length(unique(subs$class)))
print("en el Set 2")
```

```
## [1] "en el Set 2"
```

```
print(paste("en clasificacion jerarquica por aglomeracion",corteE2[length(corteE2)]))
```

```
## [1] "en clasificacion jerarquica por aglomeracion 1"
```

```
kmd <- kmeans(subs2,4)
print(paste("en K-medias por otro lado reultaria en el cluster",unique(kmd$cluster)))
```

```
## [1] "en K-medias por otro lado reultaria en el cluster 1"
## [2] "en K-medias por otro lado reultaria en el cluster 3"
## [3] "en K-medias por otro lado reultaria en el cluster 2"
## [4] "en K-medias por otro lado reultaria en el cluster 4"
```

```
#-----
subs<- subset(data,subset = data$id == 3)
subs3 <- rbind(subs[, -c(1,2,5)],c(3,3))
distanceE3 <- dist(subs3,method = "euclidean")
clusterE3 <- hclust(distanceE3,method = mejor3$method)
corteE3 <- cutree(clusterE3,k=length(unique(subs$class)))
print("en el Set 3")
```

```
## [1] "en el Set 3"
```

```

print(paste("en clasificacion jerarquica por aglomeracion",corteE3[length(corteE3)]))

## [1] "en clasificacion jerarquica por aglomeracion 1"

kmd <- kmeans(subs3,4)
print(paste("en K-medias por otro lado reultaria en el cluster",unique(kmd$cluster)))

## [1] "en K-medias por otro lado reultaria en el cluster 3"
## [2] "en K-medias por otro lado reultaria en el cluster 4"
## [3] "en K-medias por otro lado reultaria en el cluster 1"
## [4] "en K-medias por otro lado reultaria en el cluster 2"

#-----
subs<- subset(data,subset = data$id == 4)
subs4 <- rbind(subs[,-c(1,2,5)],c(3,3))
distanceE4 <- dist(subs4,method = "euclidean")
clusterE4 <- hclust(distanceE4,method = mejor4$method)
corteE4 <- cutree(clusterE4,k=length(unique(subs$class)))
print("en el Set 4")

## [1] "en el Set 4"

print(paste("en clasificacion jerarquica por aglomeracion",corteE4[length(corteE4)]))

## [1] "en clasificacion jerarquica por aglomeracion 1"

kmd <- kmeans(subs4,4)
print(paste("en K-medias por otro lado reultaria en el cluster",unique(kmd$cluster)))

## [1] "en K-medias por otro lado reultaria en el cluster 1"
## [2] "en K-medias por otro lado reultaria en el cluster 4"
## [3] "en K-medias por otro lado reultaria en el cluster 2"
## [4] "en K-medias por otro lado reultaria en el cluster 3"

#-----

```