XIAO_Lin_Solutions_lab5

Lin Xiao

September 28, 2015

1.Read in the file titled Wimbledon.csv into the variable myTennisDataD. In your read.csv() remember to set the stringsAsFactors() parameter to FALSE. Remove the first two columns of your dataset, paste the names together, and replace the row numberings with these match names.

```
myTennisDataD <- read.csv("/Users/LinXIAO/Wimbledon.csv",stringsAsFactors = F)
player_names <- paste(myTennisDataD[,1],myTennisDataD[,2],sep = " vs ")
myTennisDataD <- myTennisDataD[,-(1:2)] # Remove the first 2 columns
rownames(myTennisDataD) <- player_names # Pass the pasted names to rows
```

2. There are two columns in your dataset that contain only NA values. Remove these. For all other columns that contain missing values, replace the missing values with the median values for that column. Store this cleaned dataset into the variable iFinalTennisData

```
flag <- c()
for(i in 1:ncol(myTennisDataD)){
  if(all(is.na(myTennisDataD[,i]))){  # If all values of the column are NA
    flag <- c(flag,i)   # Mark them and skip to the next loop
    next
  }
  med <- median(myTennisDataD[,i],na.rm = T)  # Remove NAs while calculating the median
  myTennisDataD[,i][is.na(myTennisDataD[,i])] <- med
}
iFinalTennisData <- myTennisDataD[,-flag] # Delete the marked columns
```

3. Create a log-Euclidean distance matrix for your data and perform single linkage and complete linkage clustering on the data and store these in the variables singeLinkage and completeLinkage respectively.

```
d <- log(dist(iFinalTennisData)) # Log-Euclidean
singeLinkage <- hclust(d,method = "single")
completeLinkage <- hclust(d,method = "complete")
```

I will be doing the next two parts together:

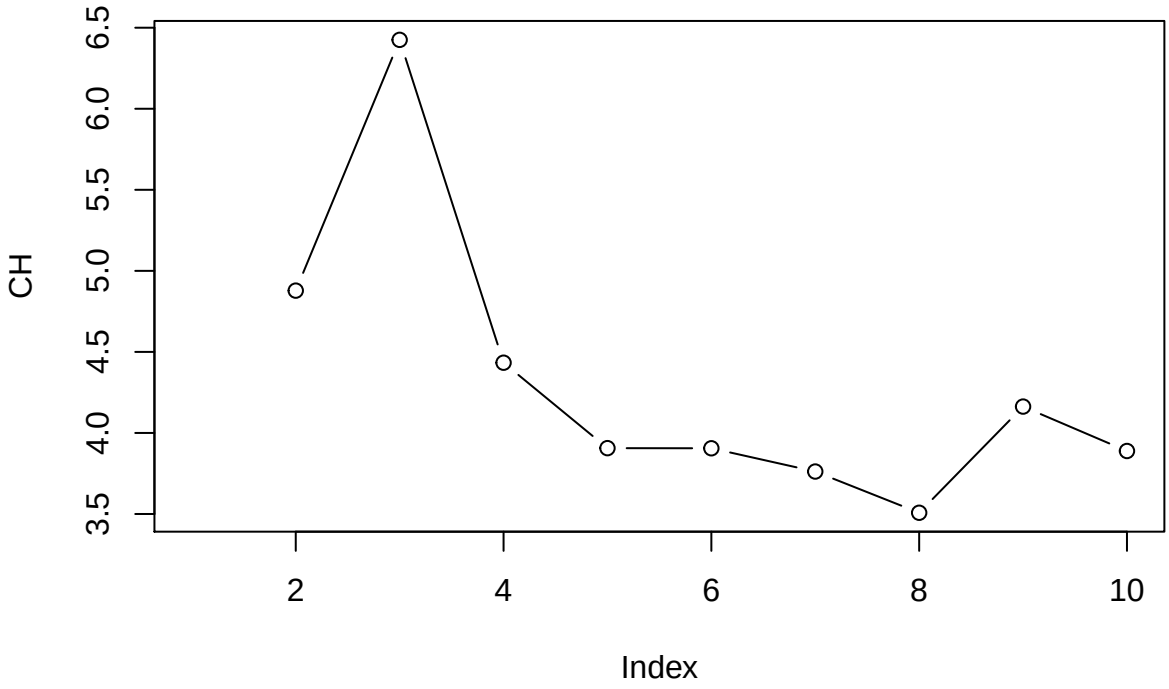
4. Using the command intCriteria() in the clusterCrit package, write a function that takes as its inputs the number of clusters to be tested (say 10), a clustered object (e.g SingleLinkage from above) and the original data frame (e.g. iFinalTennisData), and computes the CH Index for each assumed number of clusters and creates a plot of the resulting CH indices for each cluster count with a dotted vertical line indicating the maximum value. Also return the maximum CH Index computed. Test this function for up to 10 clusters with both single and complete linkage on iFinalTennisData, and include the maximum CH Index calculated and the related plots. What should the CH Index value be when the number of clusters is 1 and why?

5. For each clustering (based on linkage) what is the optimal number of clusters?

```
require(clusterCrit)
```

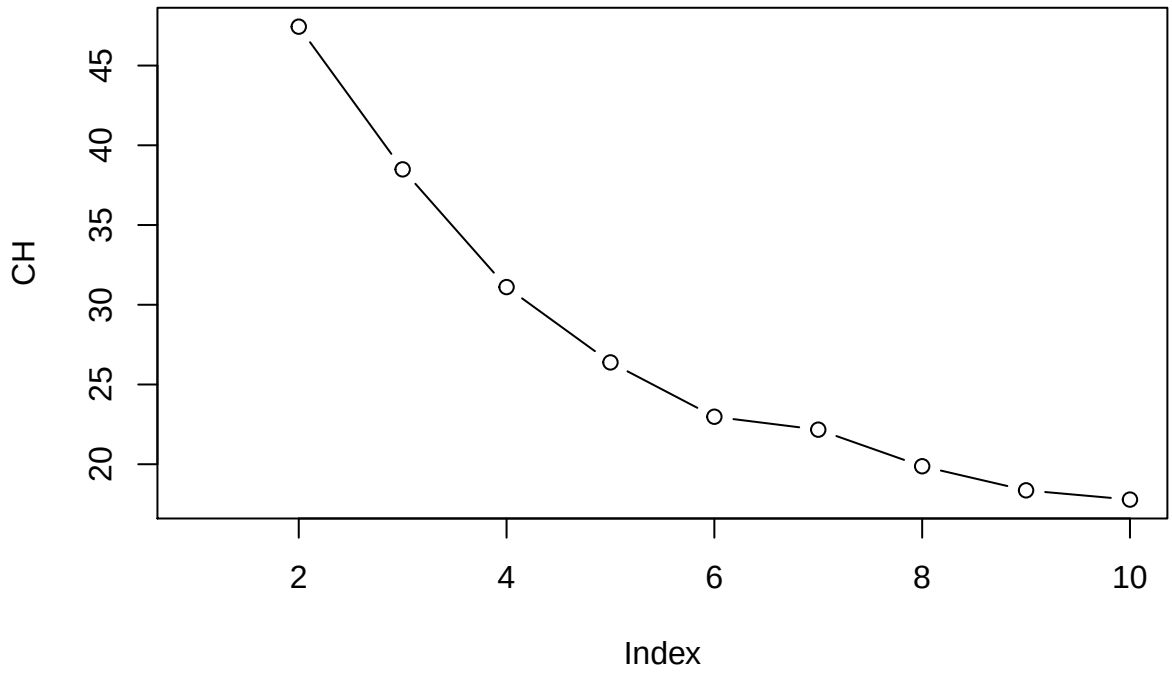
```
## Loading required package: clusterCrit
```

```
## Inputs are the number of clusters, clustered object and the clean data frame
## Outputs are the plots, the index and the value of the maximum CH
Index_CH <- function(num_cluster,cluster,datfr){
  CH <- c()
  for(i in 1:num_cluster){
    CH[i] <- intCriteria(as.matrix(datfr),cutree(cluster,k = i),crit="Calinski_Harabasz")[[1]]
  }
  plot(CH,type = "b") # Plot the CH index with respect to the number of clusters
  # Then return the index and the value of the maximum CH
  print(c("Index" = 1+which.max(CH[2:num_cluster]),"Max" = 1+max(CH[2:num_cluster])))
}
# 10 clusters with single linkage on iFinalTennisData
single_CH <- Index_CH(10,singeLinkage,iFinalTennisData)
```




```
##    Index      Max
## 3.000000 7.425097
```

```
# 10 clusters with complete linkage on iFinalTennisData
complete_CH <- Index_CH(10,completeLinkage,iFinalTennisData)
```

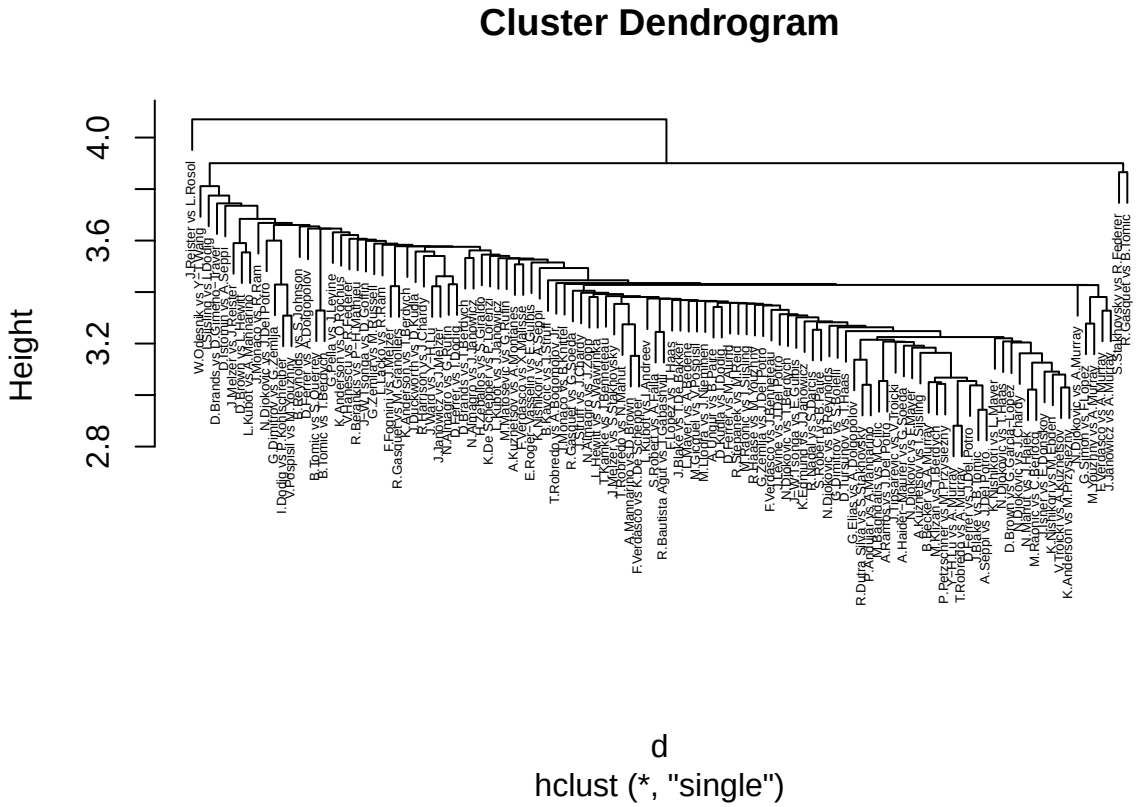



```
##   Index     Max
##  2.0000 48.4376
```

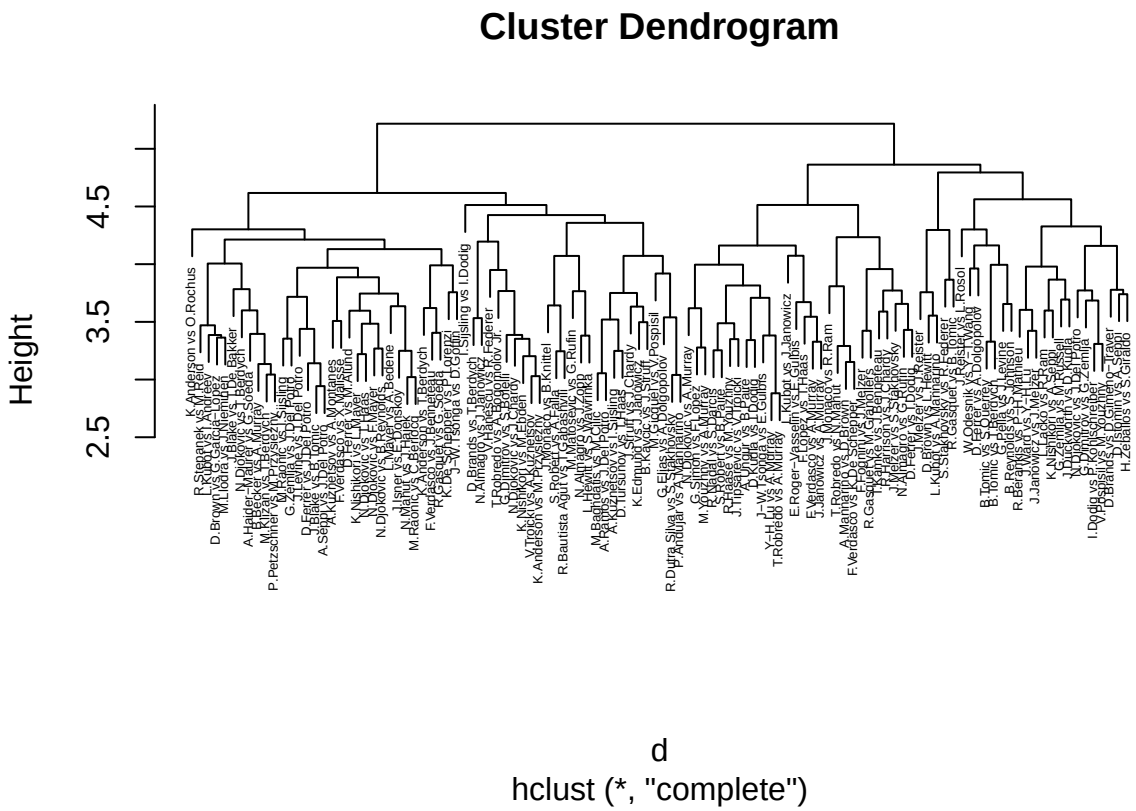
For single linkage we have maximum CH with 3 clusters and for complete linkage we have maximum CH with 2 clusters. When the number of clusters is 1, the between-clusters variable is not calculable, and the CH index is not defined, so it will generate an index which is not a number(NaN).

6. Create a dendogram for each of your two clustering assignments. Based on the data type, what is the most appropriate type of clustering and why?

```
plot(singeLinkage,cex = 0.4,hang = 1e-1)
```




```
plot(completeLinkage,cex = 0.4,hang = 1e-1)
```



Since the winner of each game will only compete with another winner who won another game, it's reasonable to have 2 clusters, and the games are more evenly distributed using complete linkage. So complete linkage is better.