Project 1

Charlotte Li

1/10/2020

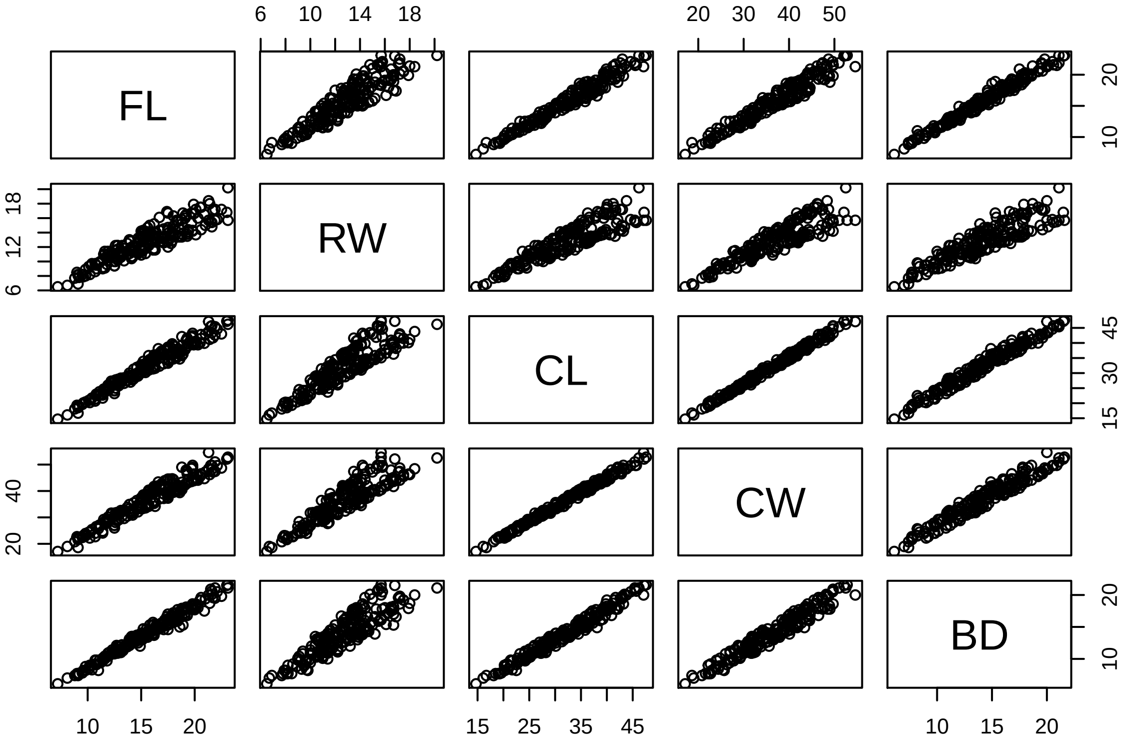
```
setwd("~/Desktop/Machine Learning/datasets")
library(foreign)
library(NbClust)
library(RcmdrMisc)
```

```
## Loading required package: car

## Loading required package: carData

## Loading required package: sandwich
```

```
crabs<-read.dta("crabs.dta")
set.seed(2011)
```

Bivariate Plots on Crabs Data's Five Features

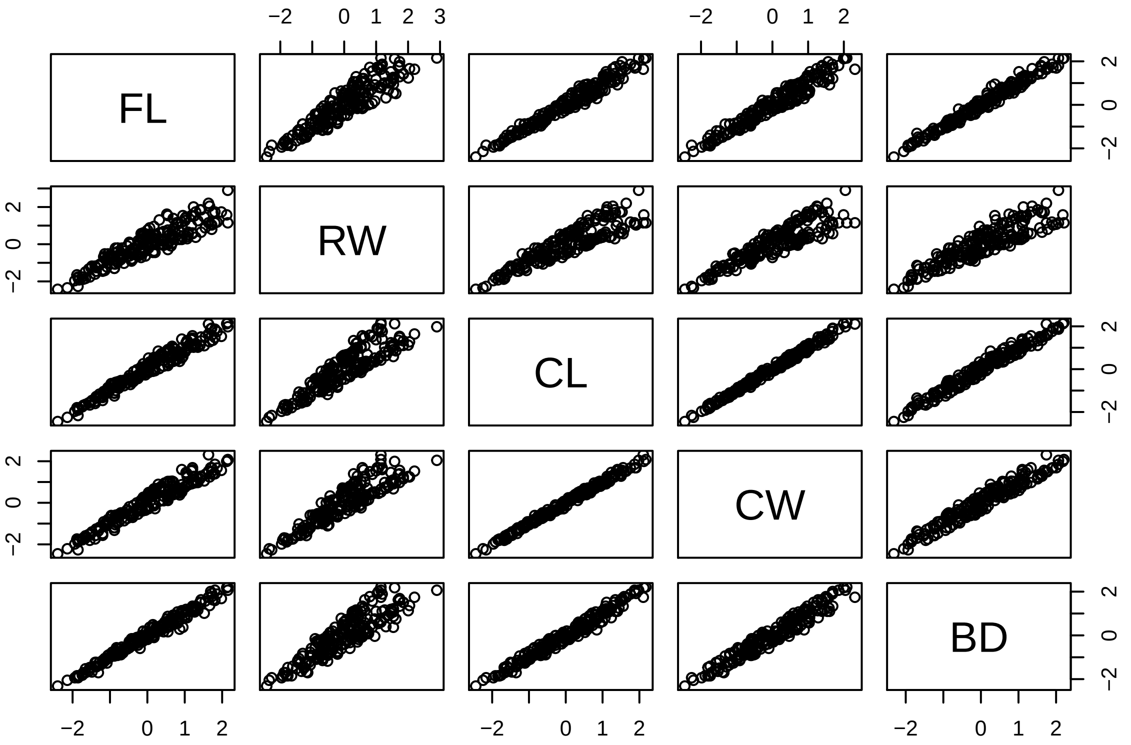
```
pairs(crabs[,4:8])
```



Looking at the above bivariable plot, I recommend rescaling rather than transforming data. First of all, scale on horizontal and vertical axis have very different scale. For example, each row and column has very different measurement range on vertical and horizontal axises. When data is rescaled the median, mean, standard deviation and variance are all rescaled by the same constant. Rescaling will change the spread of the data as well as the position of the data points. Yet, the shape of distribution and the relative attributes of the curve remain unchanged. On the other hand, transformation changes the data shape. Therefore, I would recommend rescaling.

Measurements Rescaling

```
crabs.stdz <- crabs
crabs.stdz[,4:8] <- scale(crabs[,4:8])
pairs(crabs.stdz[,4:8])
```



Generate the Principle Components

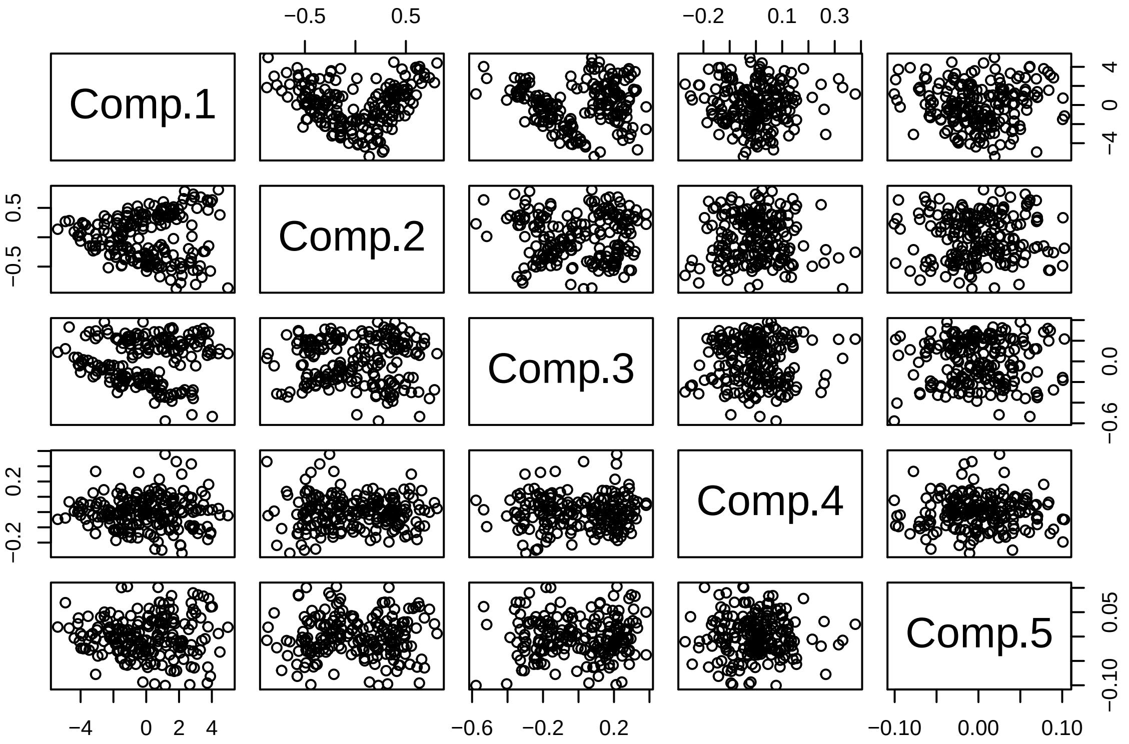
```
crabs_pc<-princomp(crabs.stdz[, 4:8],cor = T)
summary(crabs_pc)
```

```
## Importance of components:
##                          Comp.1     Comp.2      Comp.3      Comp.4       Comp.5
## Standard deviation     2.188341 0.38946785 0.215946693 0.105524202 0.0413724263
## Proportion of Variance 0.957767 0.03033704 0.009326595 0.002227071 0.0003423355
## Cumulative Proportion  0.957767 0.98810400 0.997430593 0.999657664 1.0000000000
```

```
print(crabs_pc$loadings, cutoff = 0.1)
```

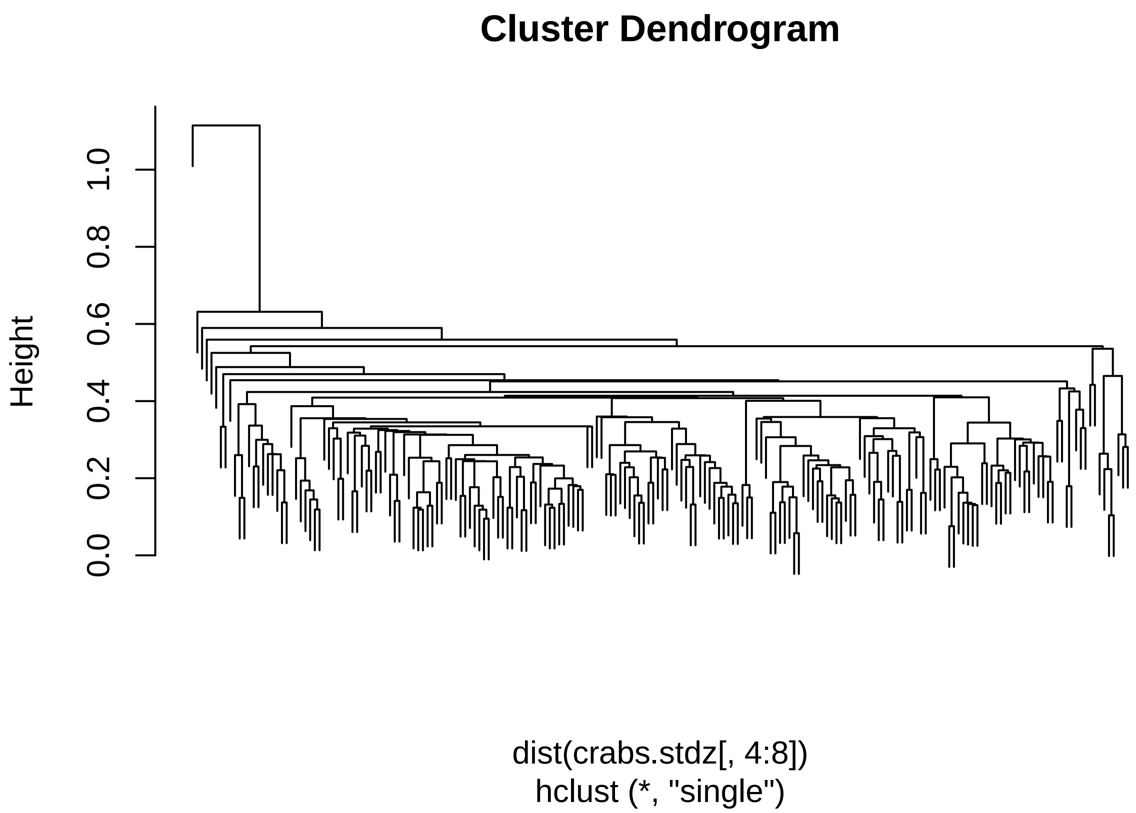
```
##
## Loadings:
##    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## FL  0.452  0.138  0.531  0.697
## RW  0.428 -0.898
## CL  0.453  0.268 -0.310        -0.792
## CW  0.451  0.181 -0.653         0.575
## BD  0.451  0.264  0.443 -0.707  0.176
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## SS loadings       1.0    1.0    1.0    1.0    1.0
## Proportion Var    0.2    0.2    0.2    0.2    0.2
## Cumulative Var    0.2    0.4    0.6    0.8    1.0
```

```
pairs(crabs_pc$scores)
```



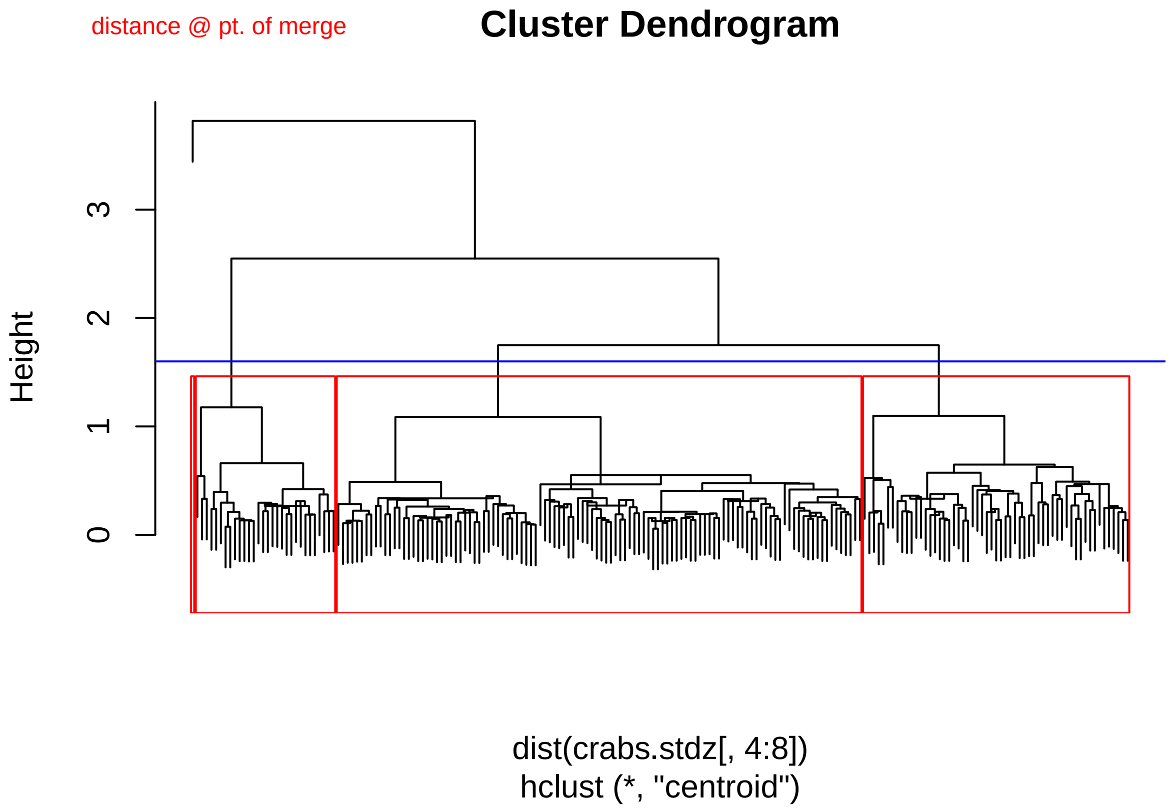
Single linkage Dendrogram

```
hcl.single <- hclust(dist(crabs.stdz[, 4:8]), meth = "single")
plot(hcl.single, labels = F)
mtext("distance @ pt. of merge", side = 2, at = 4.7, col = 2, cex = 0.75, adj = 0.25,
las = 1)
```

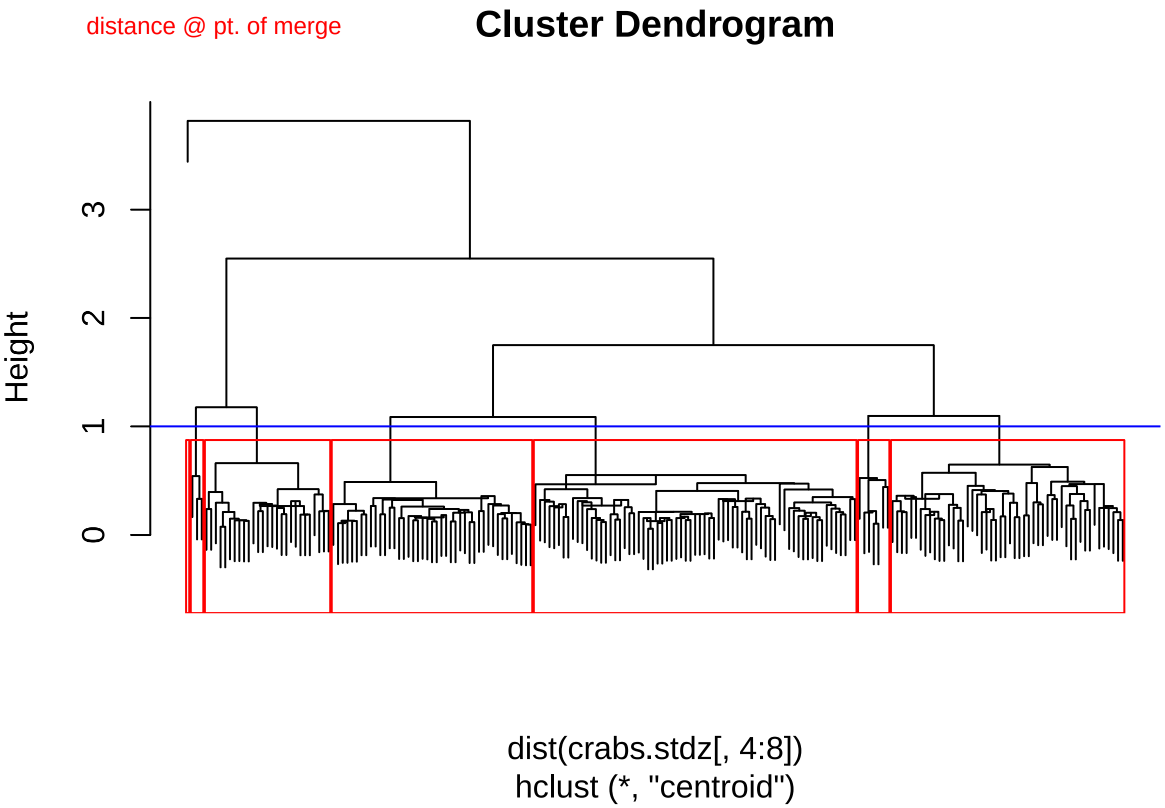


Centroid Linkage Dendrogram

```
hcl.centroid <- hclust(dist(crabs.stdz[, 4:8]), meth = "centroid")

plot(hcl.centroid, labels = F)
abline(h = 1.6, col = 4)
mtext("distance @ pt. of merge", side = 2, at = 4.7, col = 2, cex = 0.75, adj = 0.25,
las = 1)
rect.hclust(hcl.centroid, k = 4)
```



```
plot(hcl.centroid, labels = F)
abline(h = 1, col = 4)
mtext("distance @ pt. of merge", side = 2, at = 4.7, col = 2, cex = 0.75, adj = 0.25,las = 1)
rect.hclust(hcl.centroid, k = 7)
```

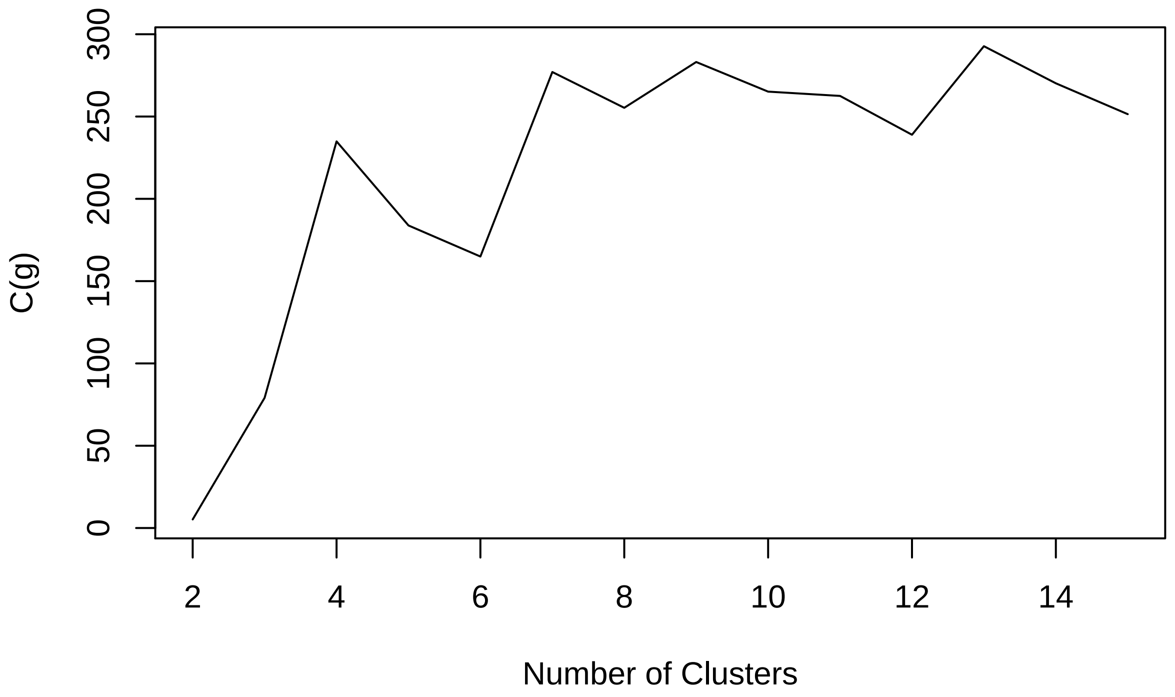


```
centroid_clu <- cutree(hcl.centroid, k = 7)
```

After comparing single and centroid linkage dendrograms, it is better to employ centroid linkage clustering method. This method defines the distance between two groups as the distance between their center. As shown on the centroid cluster dendrogram, there are clear brach paths from the top and merging paths from the bottom. Moreover, once a cutoff line is drawn, there are distinct cluster seperations. For example, in the first centroid dendrogram, my cutoff point is at 1.6. This way, there are 4 clusters: one on the far left side with a single point, one smaller chunk next to it, one larger chunk and one medium sized chunk on the right. However, the third cluster is way larger than the rest, and it contains a lot points with different height differences. Therefore, I futher draw a lower cutoff line at 1 and created 7 clusters. Seven clusters allow me to seperate my points more evenly and without too much divergent height in each clustering group.

Optimal number of clusters using Centroid Linkage Method

```
Optimal.crabs.centroid <- NbClust(data = crabs.stdz[, 4:8],
                                  distance = "euclidean",method = "centroid", index = "ch")
plot(2:15,Optimal.crabs.centroid$All.index, type = "l",
     xlab = "Number of Clusters", ylab = "C(g)")
```

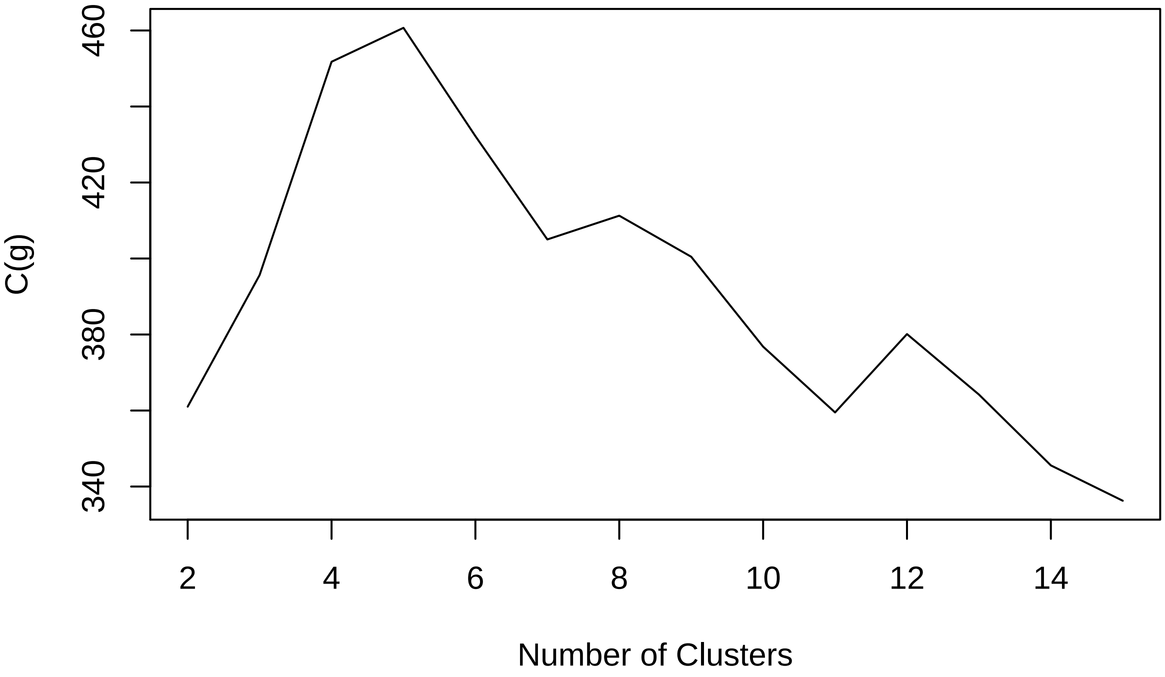


```r
## Apply Centroid Linkage Method again with k=4 clusters
optimal.centroid_clu <- cutree(hcl.centroid, k = 4)
```

Based on the plot above, the global max point is around 280 which corrosponds to 13 clusters. However, 13 clusters are too many to be desiable. Accrording to Farley and Raftery (1998), a heuristic that works well in practice is to select the number of clusters corresponding to the first decisive local maximum (if any) over all the parametrizations considered.Looking at the table again, 7-cluster is the first local maximum point.Moreover, the increase is more steep and sharp reaching 4. Therefore, 4 is the optimal cluster number.

Optimal number of clusters using Nbclust function & Kmenas Method

```r
## Using NbClust function
Optimal.crabs.K <- NbClust(data = crabs.stdz[, 4:8], distance = "euclidean",
                           method = "kmeans", index = "ch")
plot(2:15,Optimal.crabs.K$All.index, type = "l",xlab = "Number of Clusters",
     ylab = "C(g)")
```



```
Optimal.crabs.K$Best.nc
```

```
## Number_clusters     Value_Index
##          5.0000        460.6957
```

```
## Using Kmeans function
clu_km<-kmeans(crabs.stdz[,4:8],nstart = 100,5)
```

Results Comparison

```
##Compare Optimal Cluster (NbClust) and  Centroid Linkage Clustering
opt<-xtabs(~optimal.centroid_clu+Optimal.crabs.K$Best.partition)
##Label Switching to maximize diagonal
opt[,c(1,4,5,2,5)]
```

```
##                     Optimal.crabs.K$Best.partition
## optimal.centroid_clu  1  4  5  2  5
##                    1 28  2  0  0  0
##                    2  0 45  0 44  0
##                    3  0  0 29  0 29
##                    4  0  0  1  0  1
```

Based on the cross tabulation above, the maximum agreement between Kmeans method using NbClust and centroid method is the sum of the diagonal values: 28+45+29=102.

```
##Evaluate known demographics frequency distribution for the k-means cluster solution
dem_table<-xtabs(~Optimal.crabs.K$Best.partition+crabs.stdz$sex+crabs.stdz$species)
dem_table
```

```
## , , crabs.stdz$species = Blue
##
##                               crabs.stdz$sex
## Optimal.crabs.K$Best.partition Female Male
##                              1     12    9
##                                2     15   10
##                                3      6   15
##                                4     16   11
##                                5      1    5
##
## , , crabs.stdz$species = Orange
##
##                                  crabs.stdz$sex
## Optimal.crabs.K$Best.partition Female Male
##                                1      2    5
##                                2     12    7
##                                3     15   15
##                                4      6   14
##                                5     15    9
```

```
##Evaluate rowpercent table
rowPercents(dem_table)
```

```
## , , crabs.stdz$species = Blue
##
##                                  crabs.stdz$sex
## Optimal.crabs.K$Best.partition Female Male Total Count
##                                1   57.1 42.9   100    21
##                                2   60.0 40.0   100    25
##                                3   28.6 71.4   100    21
##                                4   59.3 40.7   100    27
##                                5   16.7 83.3   100     6
##
## , , crabs.stdz$species = Orange
##
##                                  crabs.stdz$sex
## Optimal.crabs.K$Best.partition Female Male Total Count
##                                1   28.6 71.4   100     7
##                                2   63.2 36.8   100    19
##                                3   50.0 50.0   100    30
##                                4   30.0 70.0   100    20
##                                5   62.5 37.5   100    24
```

Based on the above demographics (species and sex) distribution, some clusters are built well and some are not. For easier interpretation, I also made a row percent table to demostrate the distribution within each cluster. In the first table for blue crabs, cluster 3 and 5 are male dominant clusters. Cluster 3 has 71.4% male and cluster 5 has 83.3% male. In orange crabs, more clusters have very different distribution between sexes. For instance, cluster 1 and cluster 4 has a nearly 30% female and 70% male crabs. Cluster 2 has over 60% female and 36% male and similar distribution in cluster 5.

The purpose of Clustering is to create homogenous groups within hetergenous groups. In this case, cluster 3 and 5 in blue crabs are male dominant and cluster 1,2,4,5 in orange crabs are female dominant. These clusters are almost homogenous on one sex which make them good clusters. This means that these clusters seperated well between 2 genders. On the countrary, cluster 3 in orange crabs, there are equal distribution between female and male crabs meaning this cluster did not seperate male from female well.