MVA_Assignment_5

Aman

10/15/2020

Assignment 5 - Clustering

This document applies some Clustering techniques on the Heart Failure Prediction dataset. We also look at clustering validation mechanisms and then perform a profiling exercise for our clusters.

Let us load libraries and data

```
# clear environment
rm(list = ls())

# defining libraries

library(ggplot2)
library(dplyr)
library(PerformanceAnalytics)
library(data.table)
library(sqldf)
library(nortest)
library(tidyverse)
library(MASS)
library(rpart)
library(class)
library(ISLR)
library(scales)
library(ClustOfVar)
library(GGally)
library(reticulate)
library(ggthemes)
library(RColorBrewer)
library(gridExtra)
library(kableExtra)
library(Hmisc)
library(corrplot)
library(energy)
library(nnet)
library(Hotelling)
library(car)
library(devtools)
library(ggbiplot)
```

```r
library(factoextra)
library(rgl)
library(FactoMineR)
library(cluster)
library(magrittr)
library(NbClust)
```

```r
# reading data
data <- read.csv('/Users/mac/Downloads/heart_failure_clinical_records_dataset.csv')
str(data)
```

```
## 'data.frame':    299 obs. of  13 variables:
##  $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
##  $ anaemia                 : int  0 0 0 1 1 1 1 1 0 1 ...
##  $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
##  $ diabetes                : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
##  $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ platelets               : num  265000 263358 162000 210000 327000 ...
##  $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
##  $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
##  $ sex                     : int  1 1 1 1 0 1 1 1 0 1 ...
##  $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
##  $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
##  $ DEATH_EVENT             : int  1 1 1 1 1 1 1 1 1 1 ...
```

Let's scale the data for the independent variables - we will invoke dplyr now

```r
data_2 <- data[,-13] %>%
  na.omit() %>%          # Remove missing values (NA)
  scale()                # Scale variables
```

Let's assess clustering tendency of the data first

We use the visual and the hopkins statistic approach for this. With hopkins' statistic, we see how close the value is to 1 to identify if our data is actually clusterable.

The hopkins statistic is defined as -

$$H = \frac{\sum\limits_{i=1}^{m} u_i^d}{\sum\limits_{i=1}^{m} u_i^d + \sum\limits_{i=1}^{m} w_i^d}$$

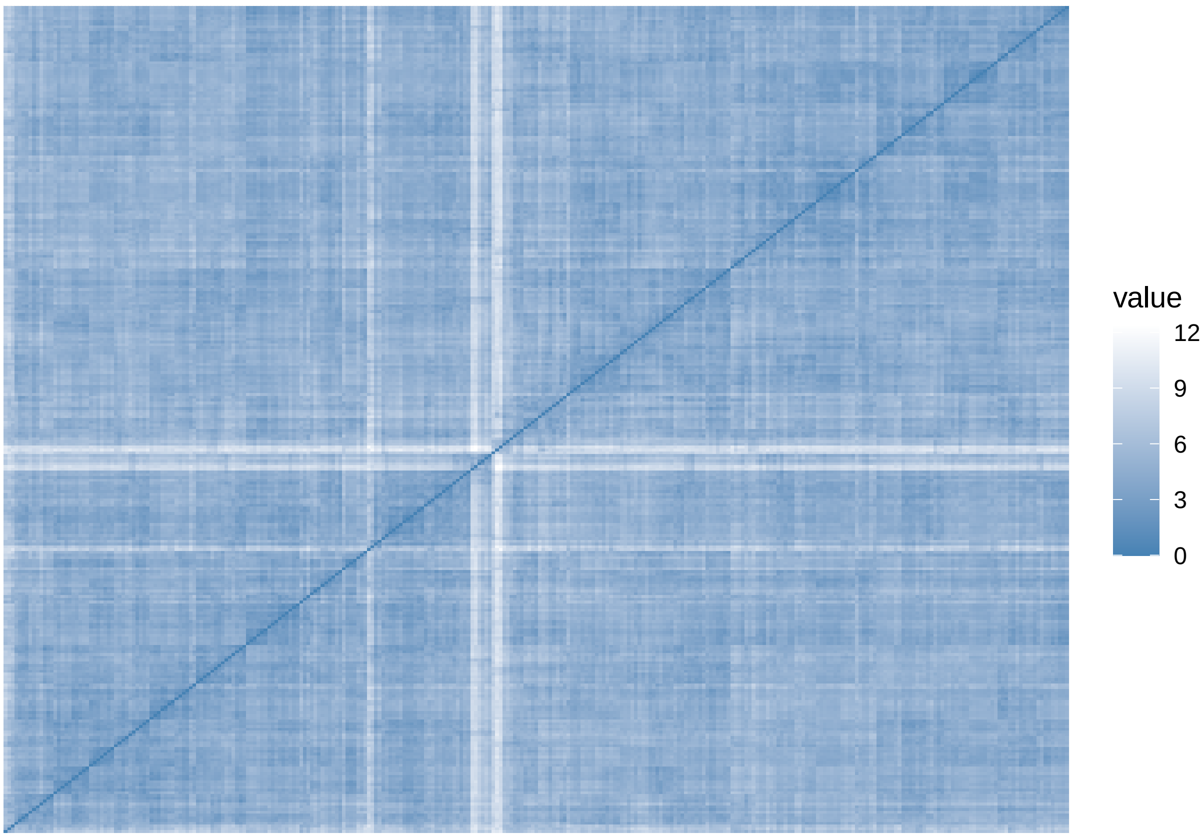
where;
X is a set of n points
Y is a set of m uniformly randomly distributed data points
$m << n$ is a sample from a set of n data points
u_i is the distance of y_i from it's nearest neighbours in X

w_i is the distance of m randomly chosen x_i from it's nearest neighbours in X

A value close to 1 indicates data is highly clustered whereas values around 0.5 indicates random data

```
gradient.color <- list(low = "steelblue",  high = "white")
data_2 %>% get_clust_tendency(n = 50, gradient = gradient.color)
```

```
## $hopkins_stat
## [1] 0.7224523
##
## $plot
```

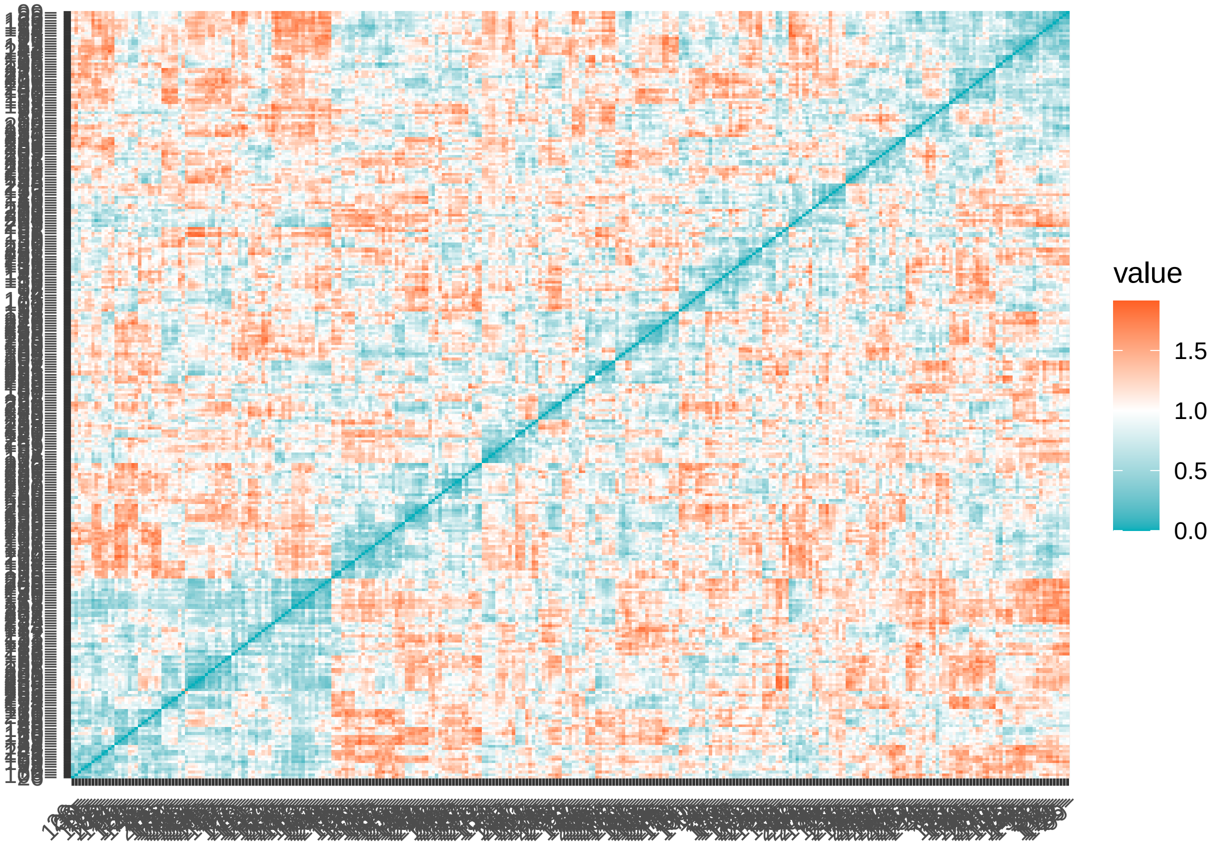


Our hopkin's statistic is 0.72. This value is still above 0.5 but no as close to 1 as we'd like. In the visual approach we cannot really see dark boxes along the diagonal as well.

Partitioning Cluster methods

Let's compute distance matrix between rows of our heart failure clinical data

```r
res.dist <- get_dist(data_2, stand = TRUE, method = "pearson")

fviz_dist(res.dist,
          gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



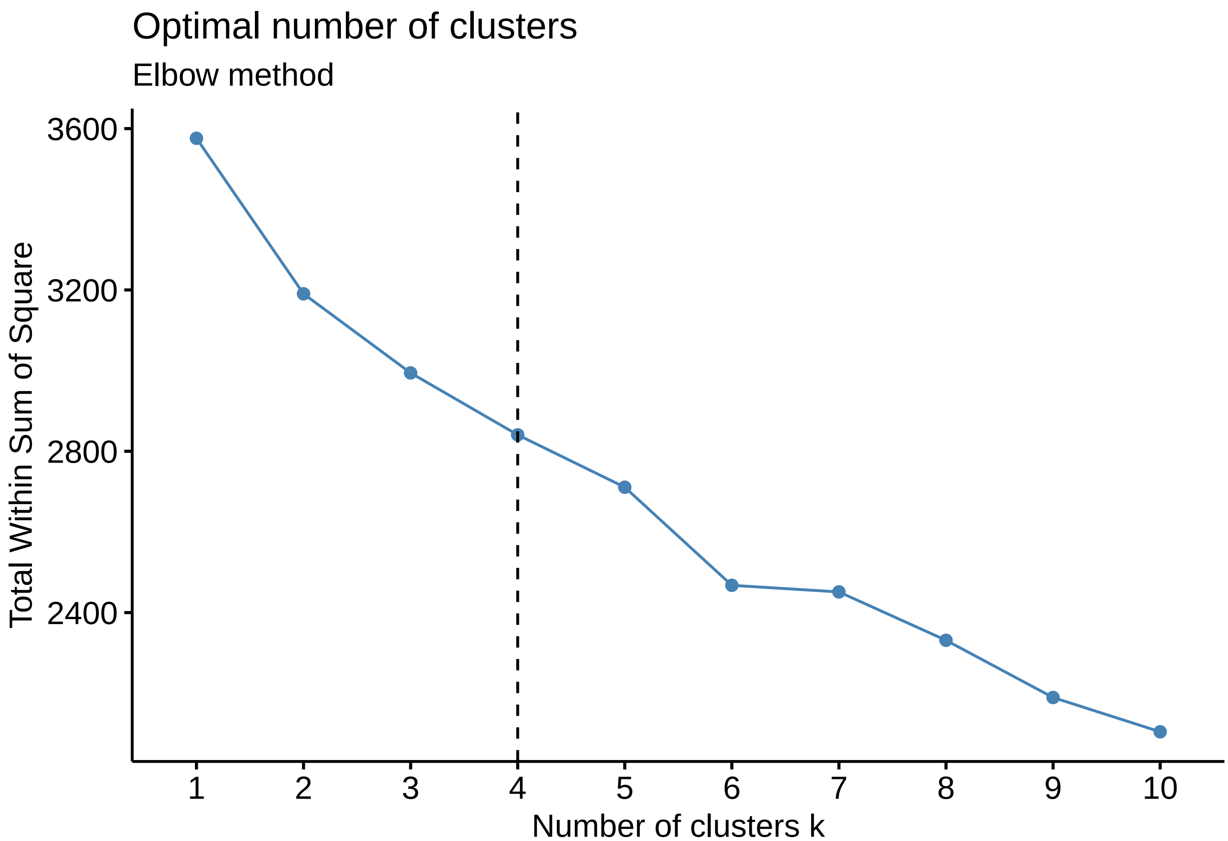
It is very hard to make sense of distances between patients with this method. We will have to try a logic or come up with groups so the clustering methods we apply on our data make sense

Let's however try k-means and explore 4 methods of choosing the optimal clusters

k - means

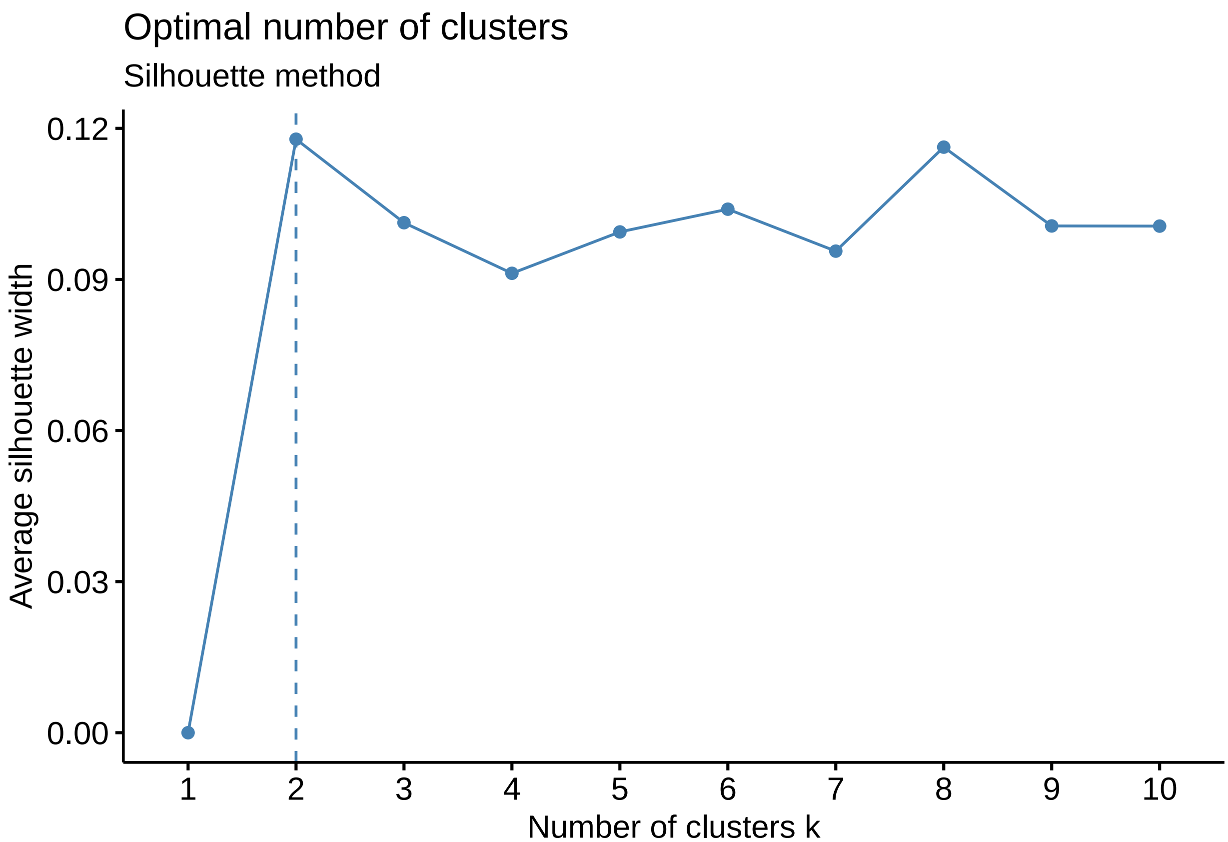
Elbow method

```
fviz_nbclust(data_2, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```



Silhouette method

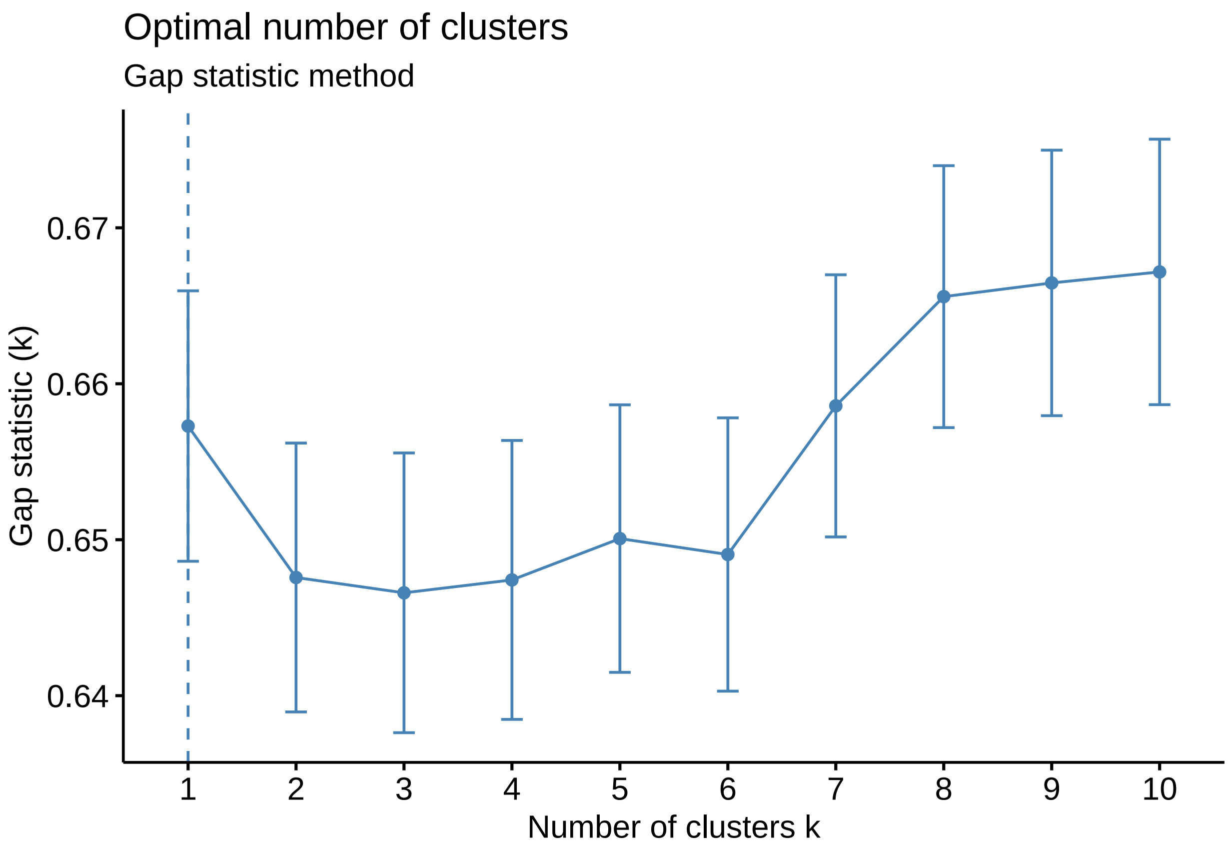
```
fviz_nbclust(data_2, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")
```



Gap statistic method

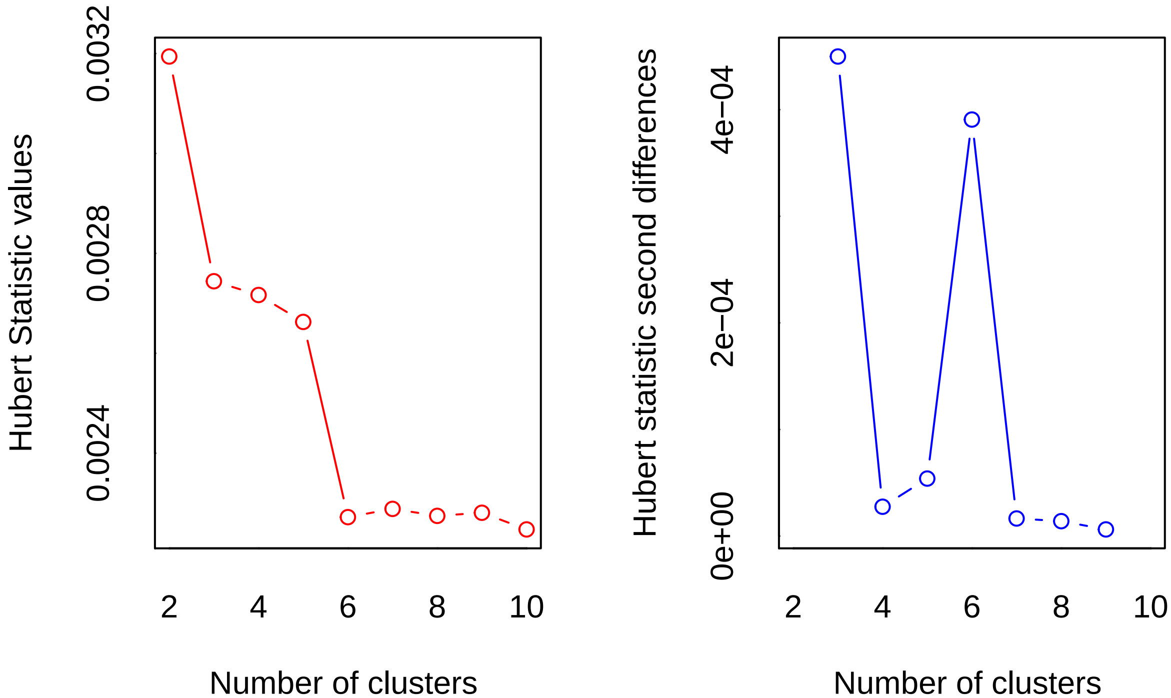
```
set.seed(123)
fviz_nbclust(data_2, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")
```

```
## Warning: did not converge in 10 iterations
```

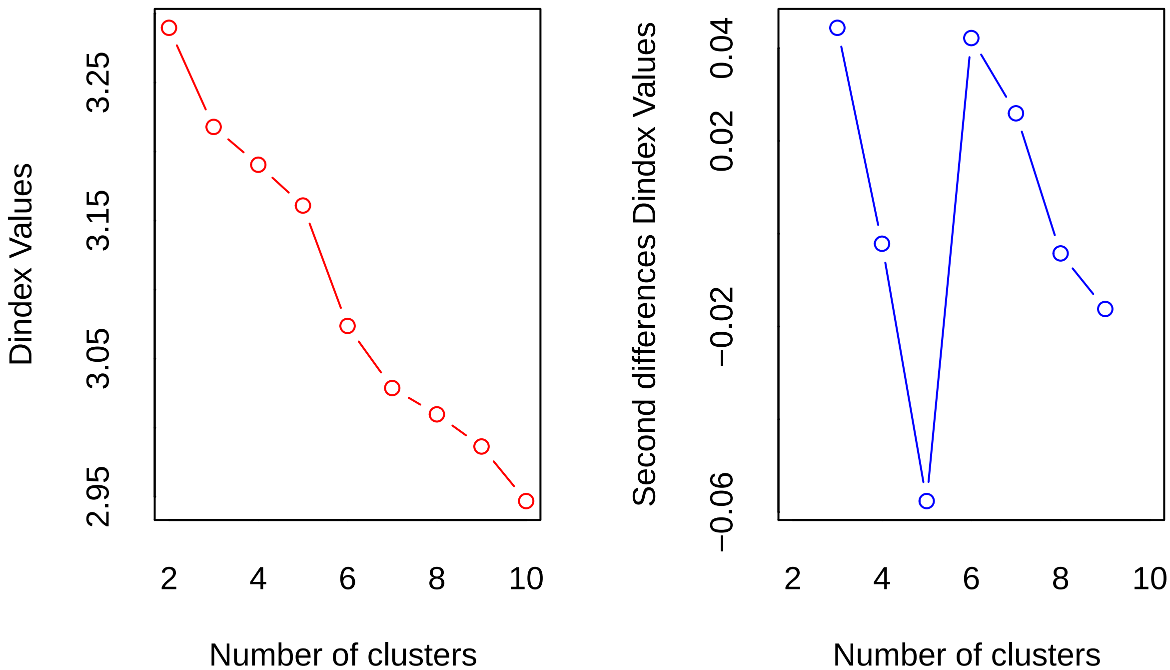


Nbclust method

```
res.nbclust <-  data[,-13] %>%
  scale() %>%
  NbClust(distance = "euclidean",
          min.nc = 2, max.nc = 10,
          method = "complete", index ="all")
```




```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```

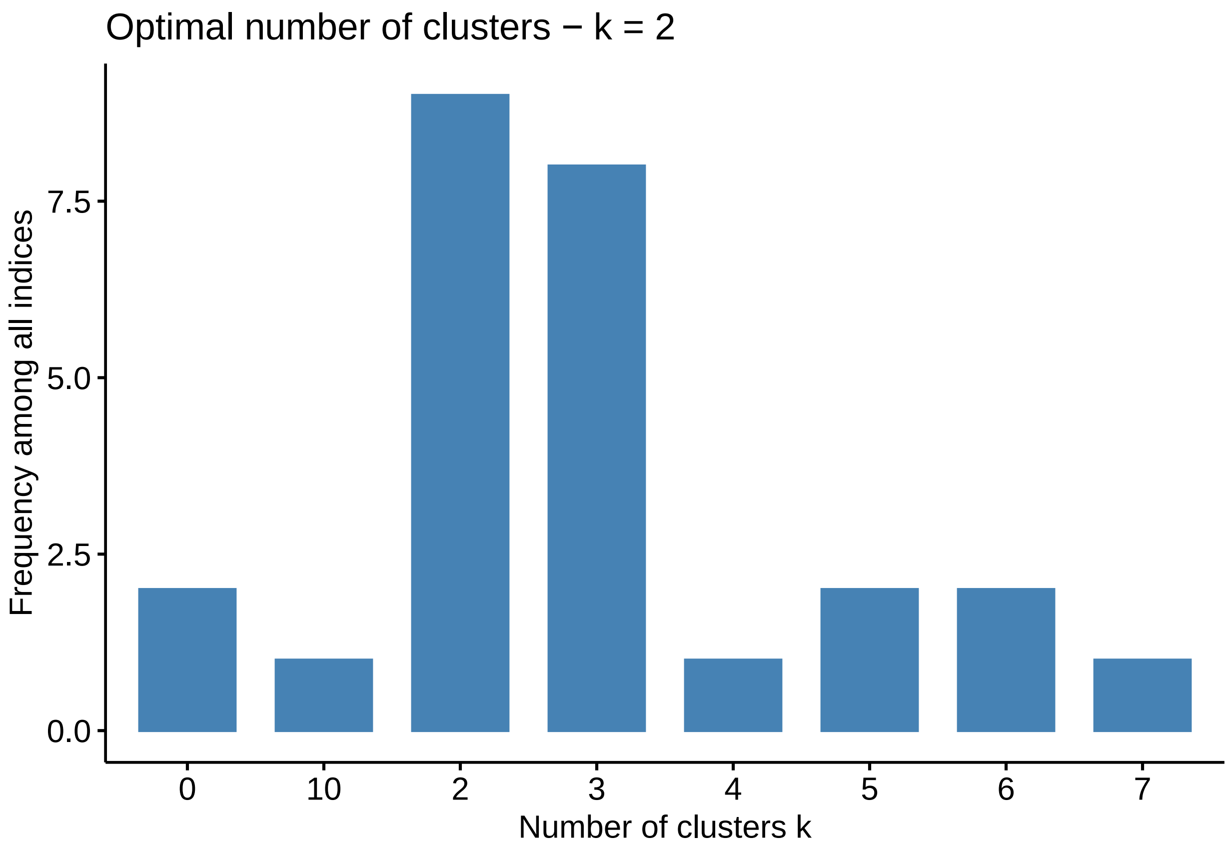



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
```

```
##
## *******************************************************************
## * Among all indices:
## * 9 proposed 2 as the best number of clusters
## * 8 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 2 proposed 5 as the best number of clusters
## * 2 proposed 6 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 1 proposed 10 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

```r
fviz_nbclust(res.nbclust, ggtheme = theme_minimal())
```

```
## Among all indices:
## ===================
## * 2 proposed  0 as the best number of clusters
## * 9 proposed  2 as the best number of clusters
## * 8 proposed  3 as the best number of clusters
## * 1 proposed  4 as the best number of clusters
## * 2 proposed  5 as the best number of clusters
## * 2 proposed  6 as the best number of clusters
## * 1 proposed  7 as the best number of clusters
## * 1 proposed  10 as the best number of clusters
##
## Conclusion
## =========================
## * According to the majority rule, the best number of clusters is  2 .
```



We see elbow plot gives 4 optimal clusters.
Silhouettes gives 2 optimal clusters.
Gap-statistic doesn't converge and finally Nbclust gives 2 optimal clusters as well.

It is important to understand that our data may not be clusterable

Note- Some theory behind gap

The gap statistic method is used in addition to elbow plot to determine the optimal number of clusters in a data. The ideal point in gap is where gap statistic is maximised. We can see 8 as optimal cluster in graph but we also use the 1-standard error rule. Choosing the cluster size to be the smallest k such that Gap(k) >= Gap (K+1) - s(k+1). Choosing this criteria we see that happens at k=1 itself. This is Gap's way of saying that the data (atleast in this form) should not be clustered.

Note - Some theory behind Si

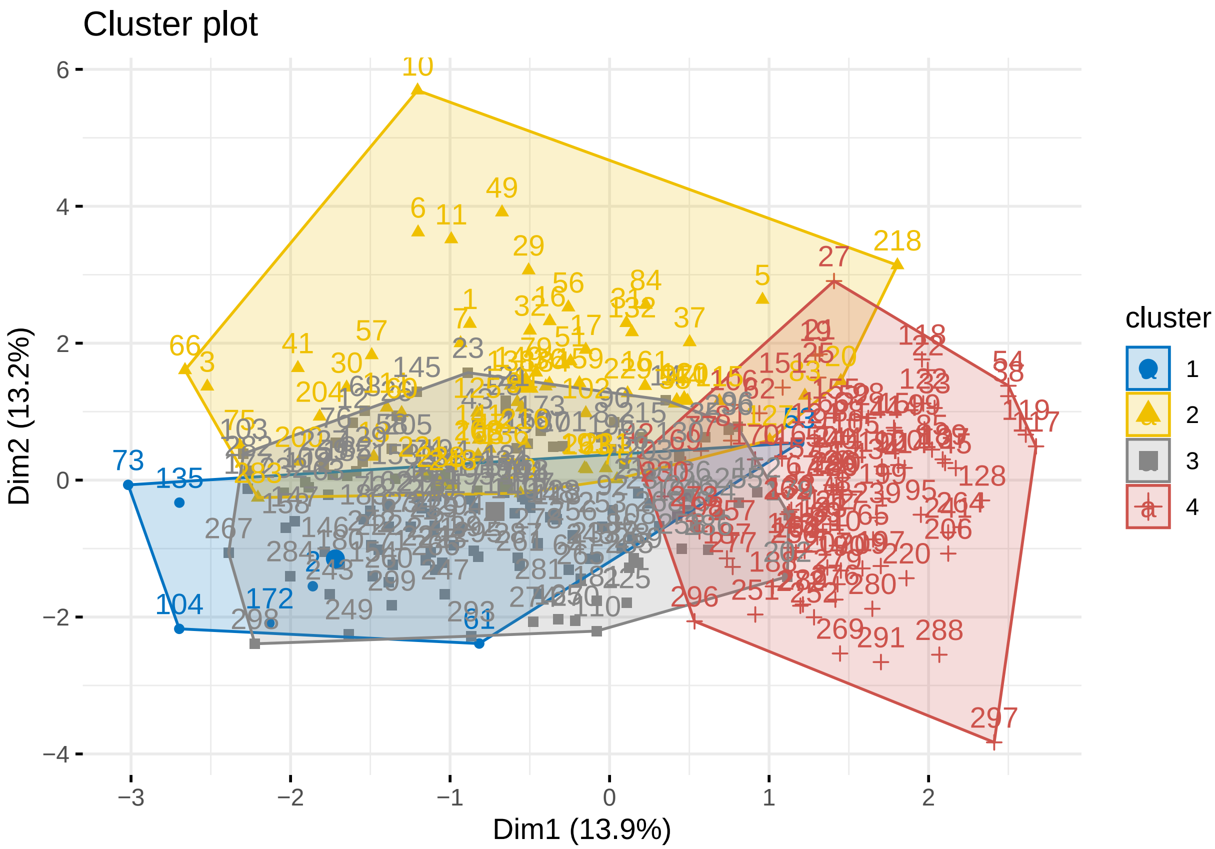
We use the silhouette coefficient to determine how good clustering is really. The silhouette coefficient measures how similar an object i is to the other objects in its own cluster versus those in the neighbor cluster. It ranges from -1 to 1.

Given the above, it seems clear that the methods are unable to give a robust cluster solution

For exposition, however let's try the solution for 4 clusters

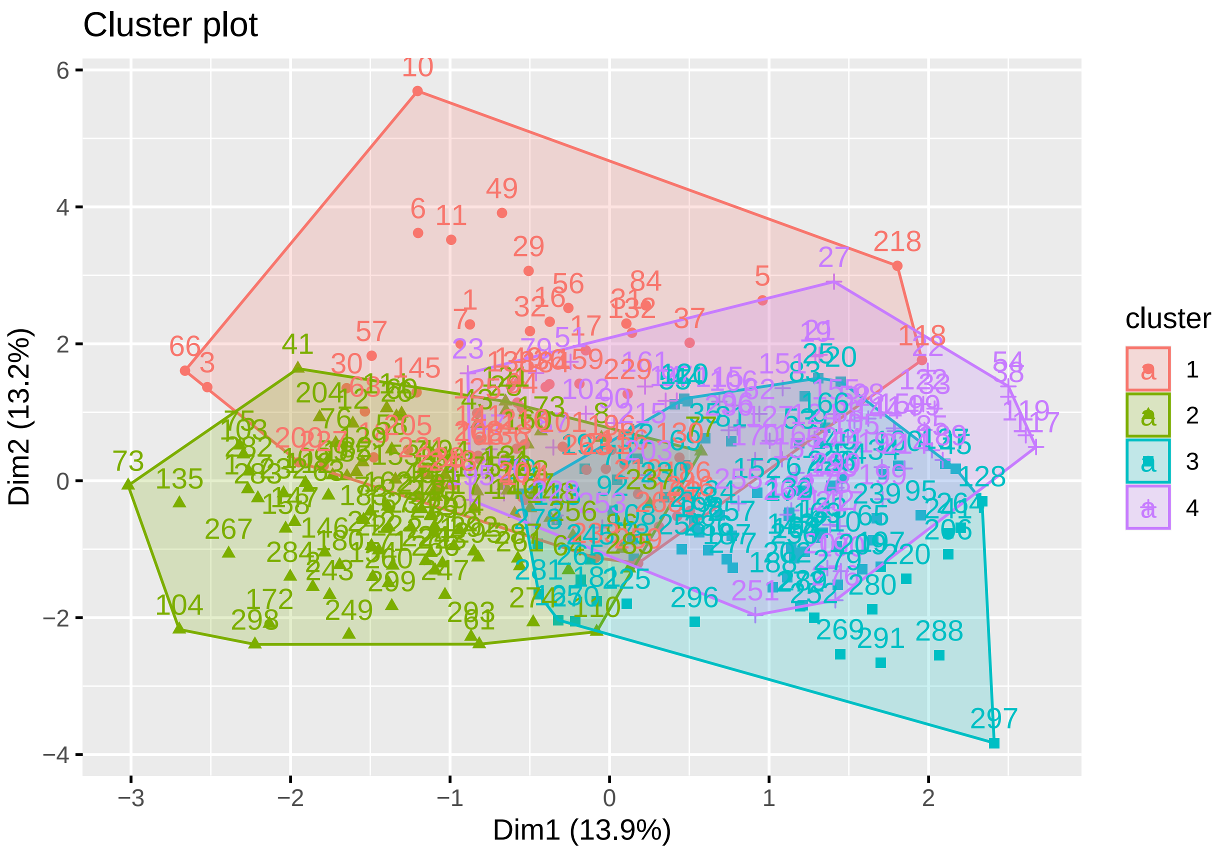
K-means

```
set.seed(123)
km.res <- kmeans(data_2, 4, nstart = 25)
fviz_cluster(km.res, data = data_2,
             ellipse.type = "convex",
             palette = "jco",
             ggtheme = theme_minimal())
```



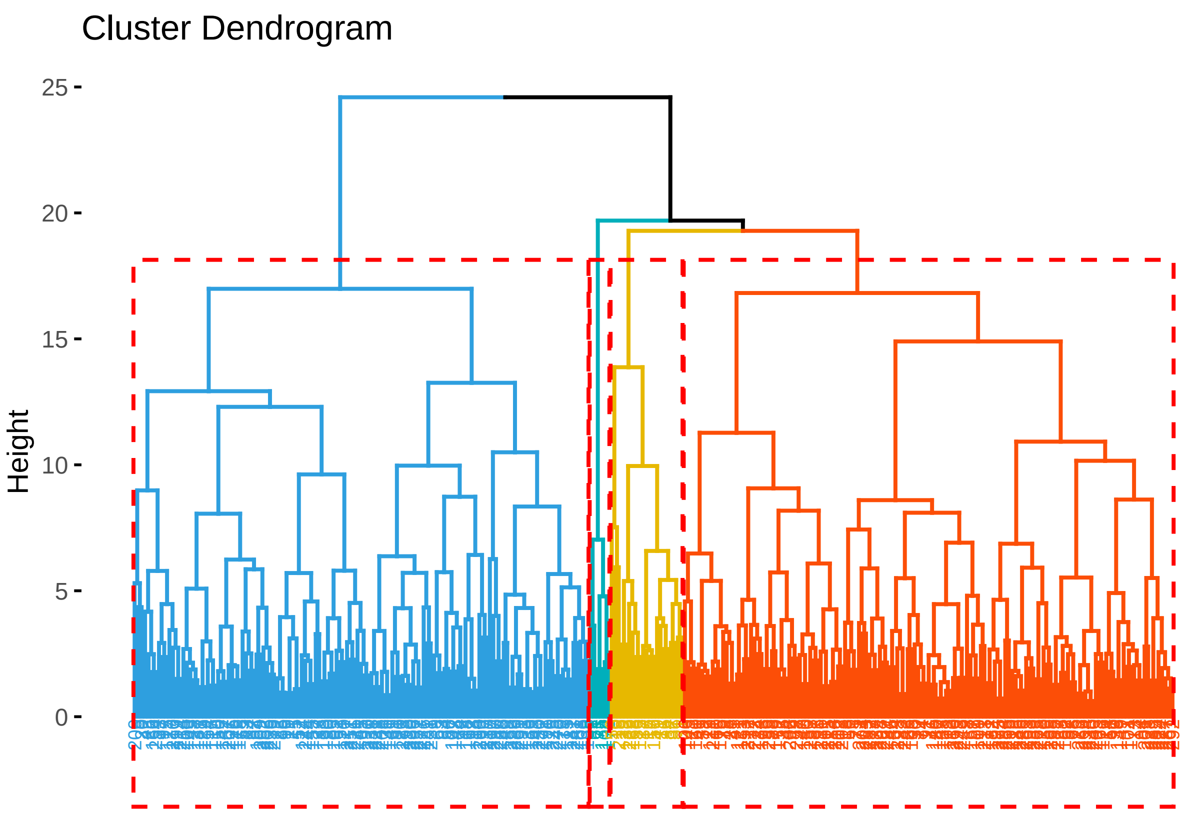
Compute PAM as well - pam is more used these days as it isn't affected by outliers as much and uses medoids

```
pam.res <- pam(data_2, 4)
fviz_cluster(pam.res)
```


Cluster plot
Dim2 (13.2%)
Dim1 (13.9%)
cluster
1
2
3
4

Heirarchical Cluster methods

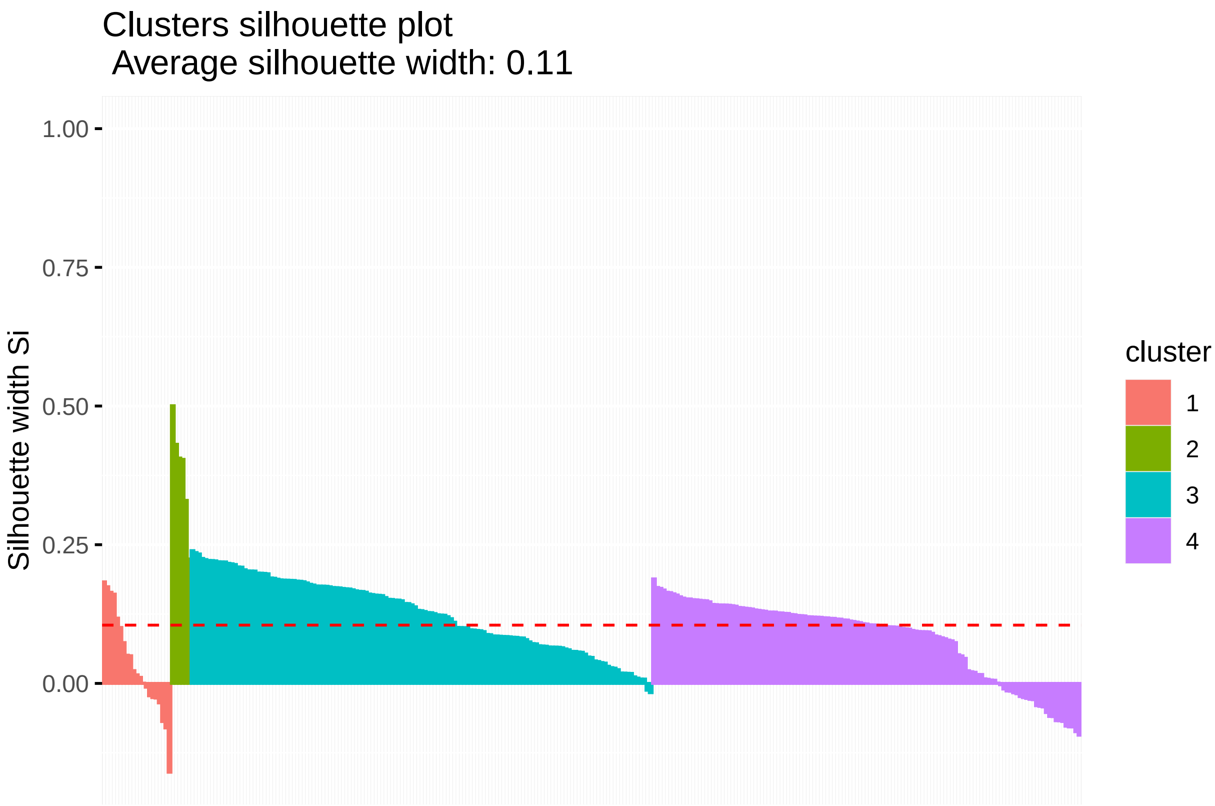
```
set.seed(123)
# Enhanced hierarchical clustering, cut in 4 groups
res.hc <- data[, -13] %>%
  scale() %>%
  eclust("hclust", k = 4, graph = FALSE)

# Visualize with factoextra
fviz_dend(res.hc, k = 4, # Cut in four groups
          cex = 0.5, # label size
          k_colors = c("#2E9FDF", "#00AFBB", "#E7B800", "#FC4E07"),
          color_labels_by_k = TRUE, # color labels by groups
          rect = TRUE, # Add rectangle around groups
          rect_border = "red" # add rect border
)
```



Let's also plot the silhouette

```
fviz_silhouette(res.hc)
```

```
##   cluster size ave.sil.width
## 1       1   21          0.03
## 2       2    6          0.38
## 3       3  141          0.13
## 4       4  131          0.08
```

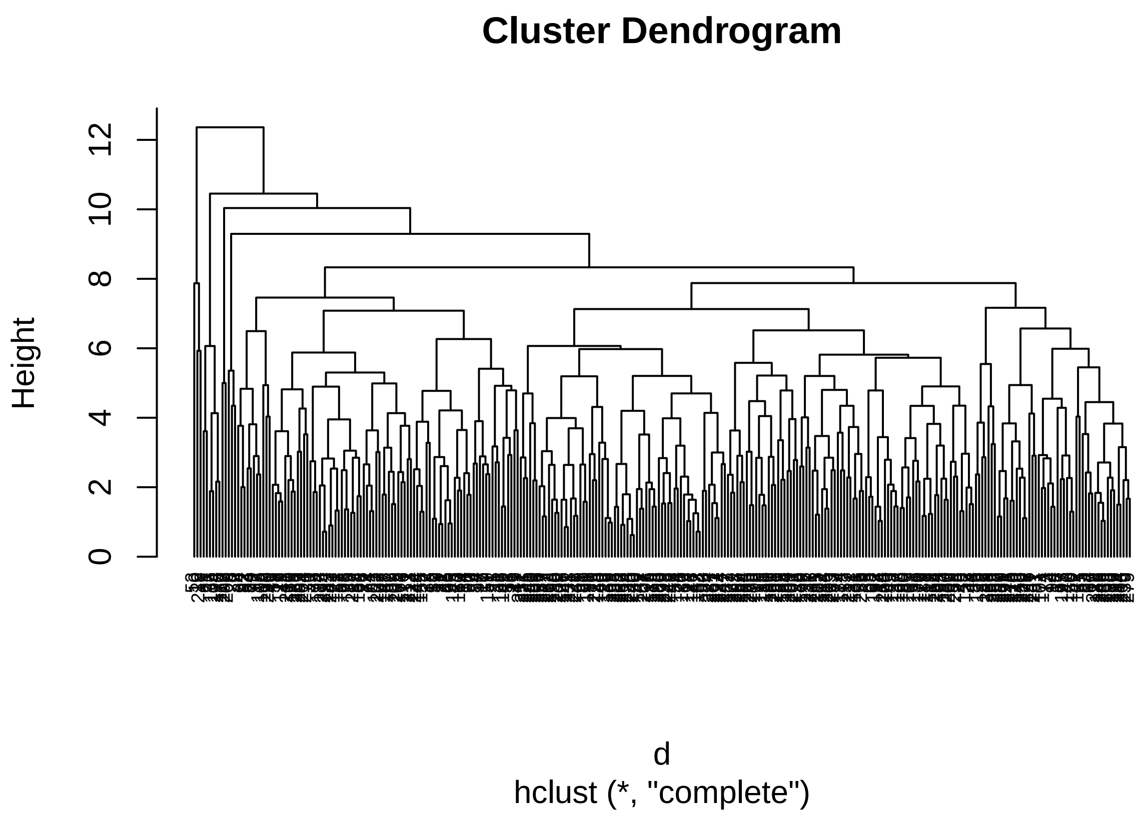


We note the average silhouette coefficients in Cluster 2 with 6 observation is 0.38 but none of the clusters have values closer to 1 indicating that the solution isn't robust as distance within cluster isn't as different as any other neighboring points.

Other Hierarchical clustering methods

Dissimilarity matrix

```
d <- dist(data_2, method = "euclidean")
# Hierarchical clustering using Complete Linkage
hc1 <- hclust(d, method = "complete" )
# Plot the obtained dendrogram
plot(hc1, cex = 0.6, hang = -1)
```

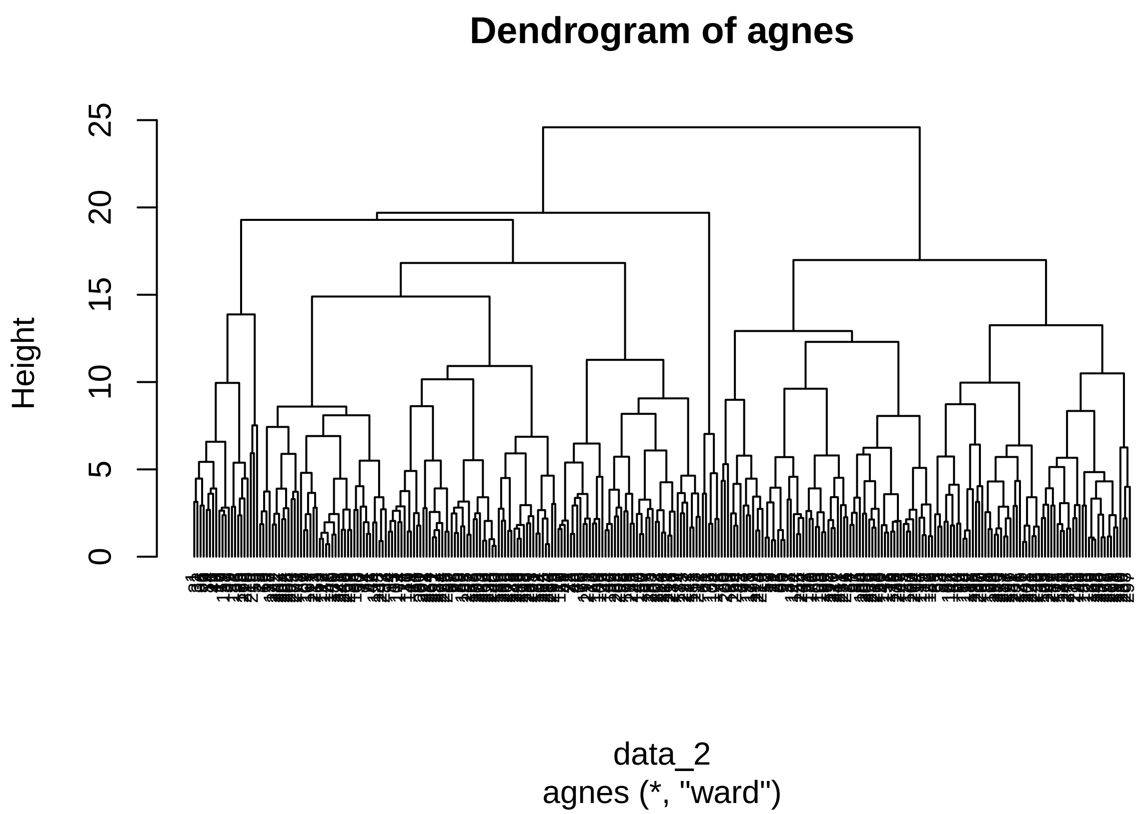


all 4 methods to assess

```
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")

# function to compute coefficient
ac <- function(x) {
  agnes(data_2, method = x)$ac
}
map_dbl(m, ac)
```

```
##   average    single  complete      ward
## 0.7308759 0.6647346 0.8275158 0.9134208
```

```
hc3 <- agnes(data_2, method = "ward")
pltree(hc3, cex = 0.6, hang = -1, main = "Dendrogram of agnes")
```



Overall, We notice ambiguity in results for clustering. While hopkins suggest data is clusterable, we do not actually obtain a good optimal number of clusters as both gap statistic and silhouette plots don't really give us good info. on what the number is and hence we don't have good validation of our clustering results

But of course, we can still profile our clusters and see if they make some sense in this dataset

Profiling the results

```r
# K-Means Cluster Analysis
fit <- kmeans(data_2, 4) # 4 cluster solution
# get cluster means
aggregate(data_2,by=list(fit$cluster),FUN=mean)
```

```
##   Group.1         age    anaemia creatinine_phosphokinase    diabetes
## 1       1 -0.23862451 -0.2976258                0.5302812 -0.05296509
## 2       2 -0.05062592 -0.1813614               -0.1851957 -0.15503318
## 3       3  0.90182890  0.6052505               -0.1789868 -0.25376570
## 4       4 -0.14868084  0.1184380               -0.1638853  0.26506403
##   ejection_fraction high_blood_pressure  platelets serum_creatinine
## 1       -0.29823451         -0.28225075 -0.2301348       -0.1652054
## 2       -0.08434809         -0.12232418  0.0322775       -0.1994361
## 3       -0.08331765          0.08172006 -0.2365614        1.3354015
## 4        0.31766589          0.27026094  0.2361004       -0.2565932
##   serum_sodium        sex    smoking        time
## 1   0.08182896  0.6779311 -0.5420541  0.34712502
## 2   0.05448993  0.7344569  1.4517270  0.07850095
## 3  -0.74977041  0.2243463 -0.3736152 -0.89928709
## 4   0.19820656 -1.1724566 -0.6236410  0.04653374
```

```r
# append cluster assignment
data_3 <- data.frame(data_2, fit$cluster)
data_4 <- cbind(data_3, DEATH_EVENT = data$DEATH_EVENT)
```

We can also check proportion of variance explained by 4 cluster solution

```r
perc.var.4 <- round(100*(1 - fit$betweenss/fit$totss),1)
names(perc.var.4) <- "Perc. 4 clus"
perc.var.4
```

```
## Perc. 4 clus
##         78.6
```

We create orginal data with assignment

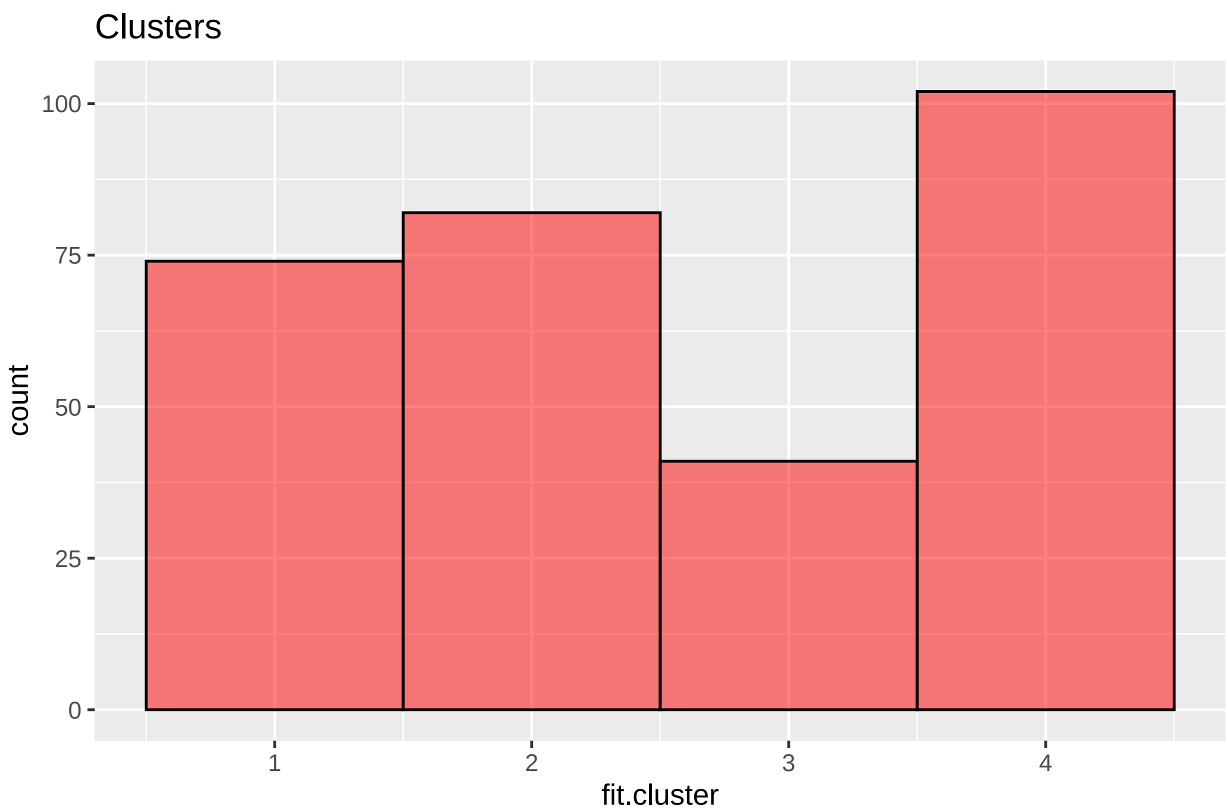
```r
org_data <- as.data.frame(t(apply(data_2, 1,
function(r)r*attr(data_2,'scaled:scale') + attr(data_2, 'scaled:center'))))
data_with_clus_assgn <- cbind(org_data,
fit.cluster=data_4$fit.cluster,DEATH_EVENT=data$DEATH_EVENT)

attach(data_with_clus_assgn)
```

Now we begin profiling exercise

Clusters with death event visualization

```r
ggplot(data_with_clus_assgn,aes(x = fit.cluster,
fill=DEATH_EVENT))+geom_histogram(binwidth = 1, color = "black",
```

```
        fill = "red",alpha = 0.5)+
  labs(title = "Clusters",
       caption = "Cluster assignments with DEATH_EVENT")+
  theme(plot.caption = element_text(hjust = 0.5,face = "italic"))
```



Cluster assignments with DEATH_EVENT

```
data_5 <- data_with_clus_assgn %>%
  group_by(fit.cluster) %>%
  dplyr::summarise(cnt = n(),cnt_death_event = sum(DEATH_EVENT),
                   prop_death_event = sum(DEATH_EVENT)/n())

data_5
```

```
## # A tibble: 4 x 4
##   fit.cluster   cnt cnt_death_event prop_death_event
##         <int> <int>           <int>            <dbl>
## 1           1    74              18            0.243
## 2           2    82              22            0.268
## 3           3    41              30            0.732
## 4           4   102              26            0.255
```

We note that the death proportion is highest in cluster 3 ~73% (highest percentage of 1s) whereas others have pretty much similar death proportion.

Clusters with age

```r
data$age_tr[data$age < 50 & data$age >= 40]="40-50"
data$age_tr[data$age < 60 & data$age >= 50]="50-60"
data$age_tr[data$age < 70 & data$age >= 60]="60-70"
data$age_tr[data$age < 80 & data$age >= 70]="70-80"
data$age_tr[data$age < 90 & data$age >= 80]="80-90"
data$age_tr[data$age < 100 & data$age >= 90]="90-100"

data_with_clus_assgn <- cbind(data_with_clus_assgn,age_tr=data$age_tr)

# Clusters with age and death event
table(data_with_clus_assgn$fit.cluster,data_with_clus_assgn$age_tr)
```

```
##
##     40-50 50-60 60-70 70-80 80-90 90-100
##   1    17    24    18    10     5      0
##   2     9    27    27    15     4      0
##   3     1     4    15     8     8      5
##   4    20    27    33    19     2      1
```

Numerically we see that cluster 3 is dominated by age range 60+. We also see cluster 1 dominated by individuals with age range < 60. Hence it makes sense that cluster 3 has a higher death event rate

```r
data_with_clus_assgn$DEATH_EVENT <- factor(data_with_clus_assgn$DEATH_EVENT)
data_with_clus_assgn$fit.cluster <- factor(data_with_clus_assgn$fit.cluster)
str(data_with_clus_assgn)
```

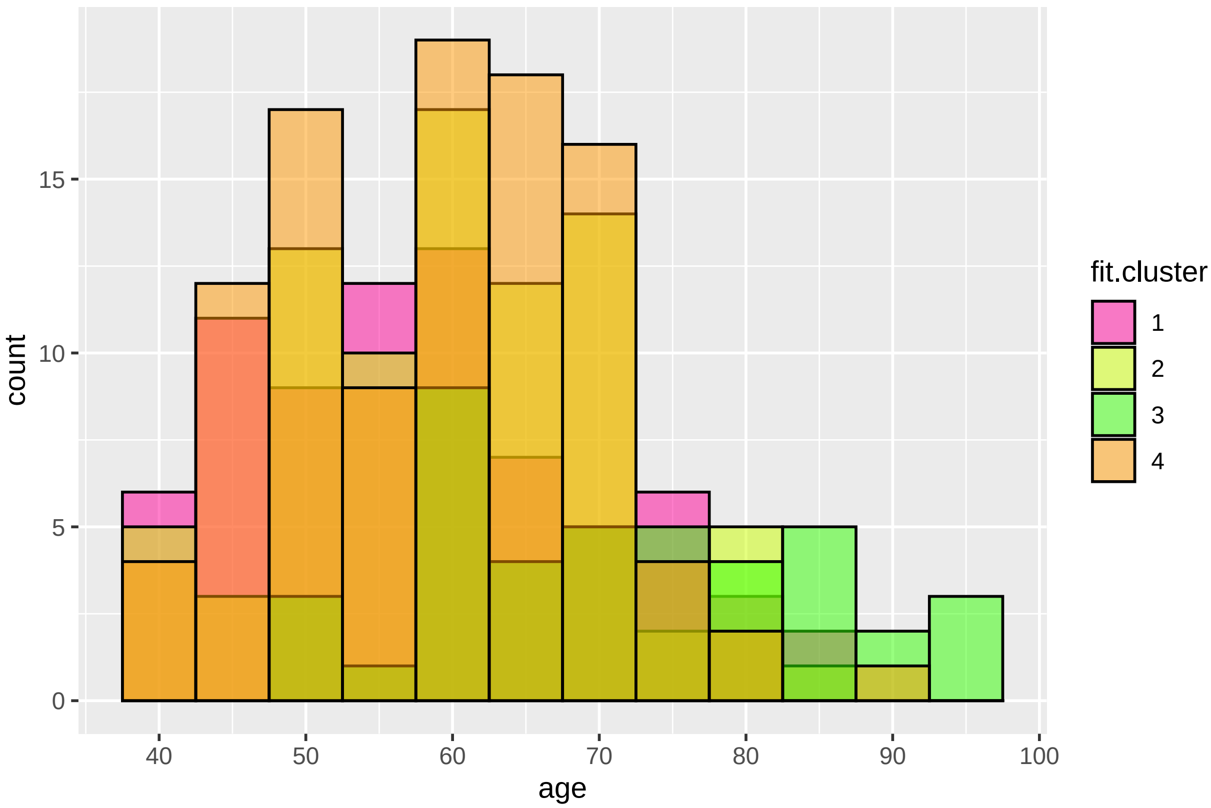
```
## 'data.frame':    299 obs. of  15 variables:
##  $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
##  $ anaemia                 : num  0 0 0 1 1 1 1 1 0 1 ...
##  $ creatinine_phosphokinase: num  582 7861 146 111 160 ...
##  $ diabetes                : num  0 0 0 0 1 0 0 1 0 0 ...
##  $ ejection_fraction       : num  20 38 20 20 20 40 15 60 65 35 ...
##  $ high_blood_pressure     : num  1 0 0 0 0 1 0 0 0 1 ...
##  $ platelets               : num  265000 263358 162000 210000 327000 ...
##  $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
##  $ serum_sodium            : num  130 136 129 137 116 132 137 131 138 133 ...
##  $ sex                     : num  1 1 1 1 0 1 1 1 0 1 ...
##  $ smoking                 : num  0 0 1 0 0 1 0 1 0 1 ...
##  $ time                    : num  4 6 7 7 8 ...
##  $ fit.cluster             : Factor w/ 4 levels "1","2","3","4": 3 1 2 3 3 3 3 2 4 3 ...
##  $ DEATH_EVENT             : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ age_tr                  : Factor w/ 6 levels "40-50","50-60",..: 4 2 3 2 3 6 4 3 3 5 ...
```

Let's check each variable with cluster assignment

Age

```r
ggplot(data_with_clus_assgn,aes(x = age,
fill = fit.cluster))+geom_histogram(binwidth = 5,
  position = "identity",
  alpha = 0.5,color = "black")+
  scale_fill_manual(values = c("#FF0099", "#CCFF00","#33FF00","#FF9900"))+
  labs(caption = "Age Distribution with Clusters")+
```

```
  theme(plot.caption = element_text(hjust = 0.5,face = "italic"))+
  scale_x_continuous(breaks = seq(40,100,10))
```

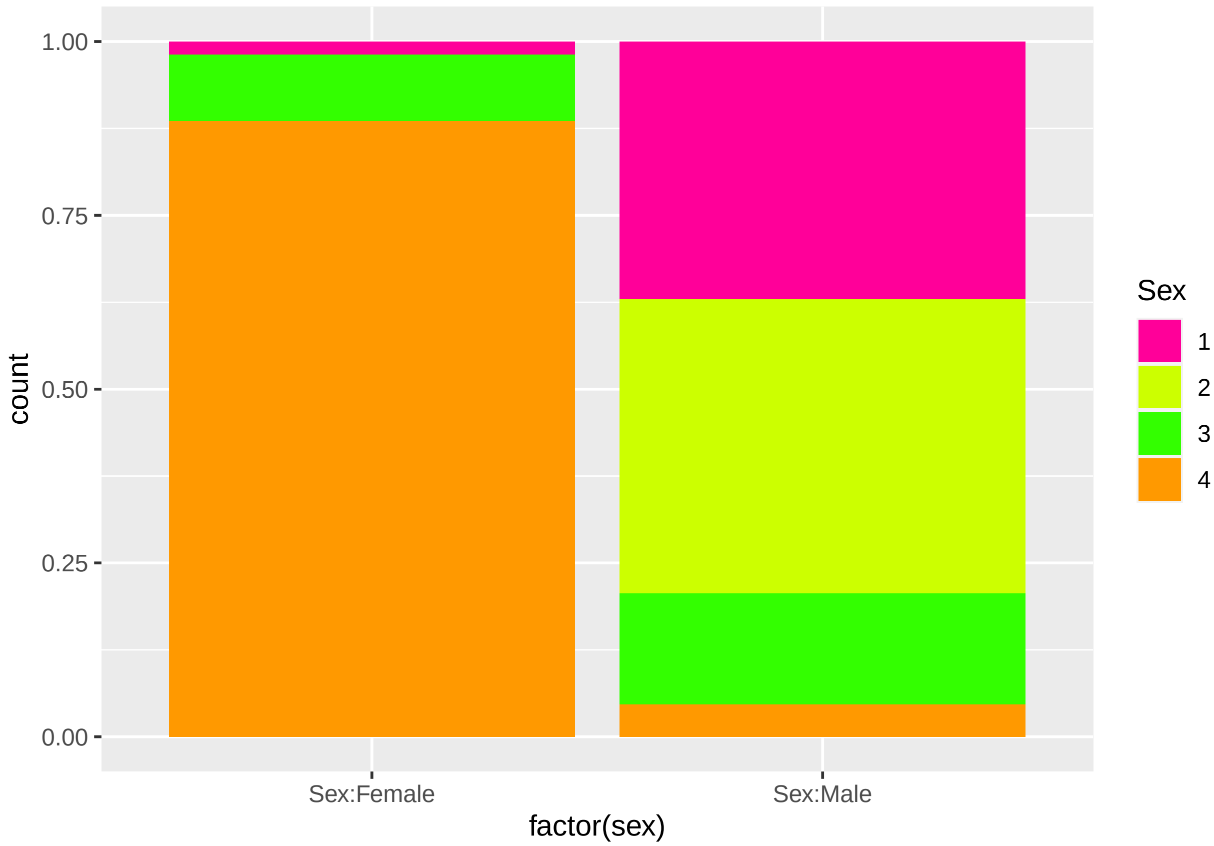


Age Distribution with Clusters

It validates our point that cluster 3 is more towards higher age groups.

Gender

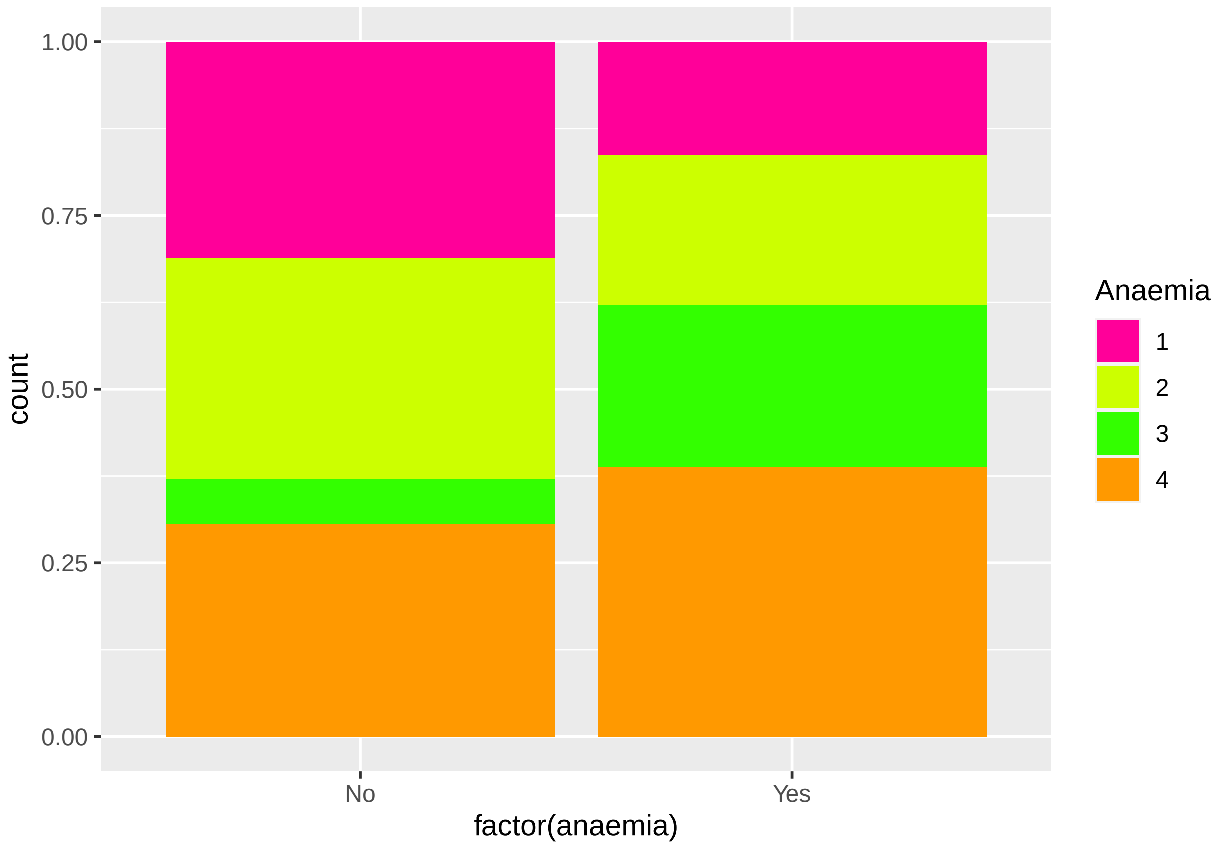
```
ggplot(data_with_clus_assgn, aes(x = factor(sex), fill = fit.cluster))+
  geom_bar(position = "fill")+
  scale_x_discrete(labels  = c("Sex:Female","Sex:Male"))+
  scale_fill_manual(values = c("#FF0099", "#CCFF00","#33FF00","#FF9900"), name = "Sex")
```



We see cluster 4 is more dominated by female population
We see cluster 1,2 is more dominated by male population (even 3 to some extent)

Anaemia

```
ggplot(data_with_clus_assgn, aes(x = factor(anaemia), fill = fit.cluster))+
  geom_bar(position = "fill")+
  scale_x_discrete(labels  = c("No","Yes"))+
  scale_fill_manual(values = c("#FF0099", "#CCFF00","#33FF00","#FF9900"), name = "Anaemia")
```



We see cluster 3 and to some extent cluster 4 have higher proprotion of anaemic individuals

Creatinine_phosphokinase

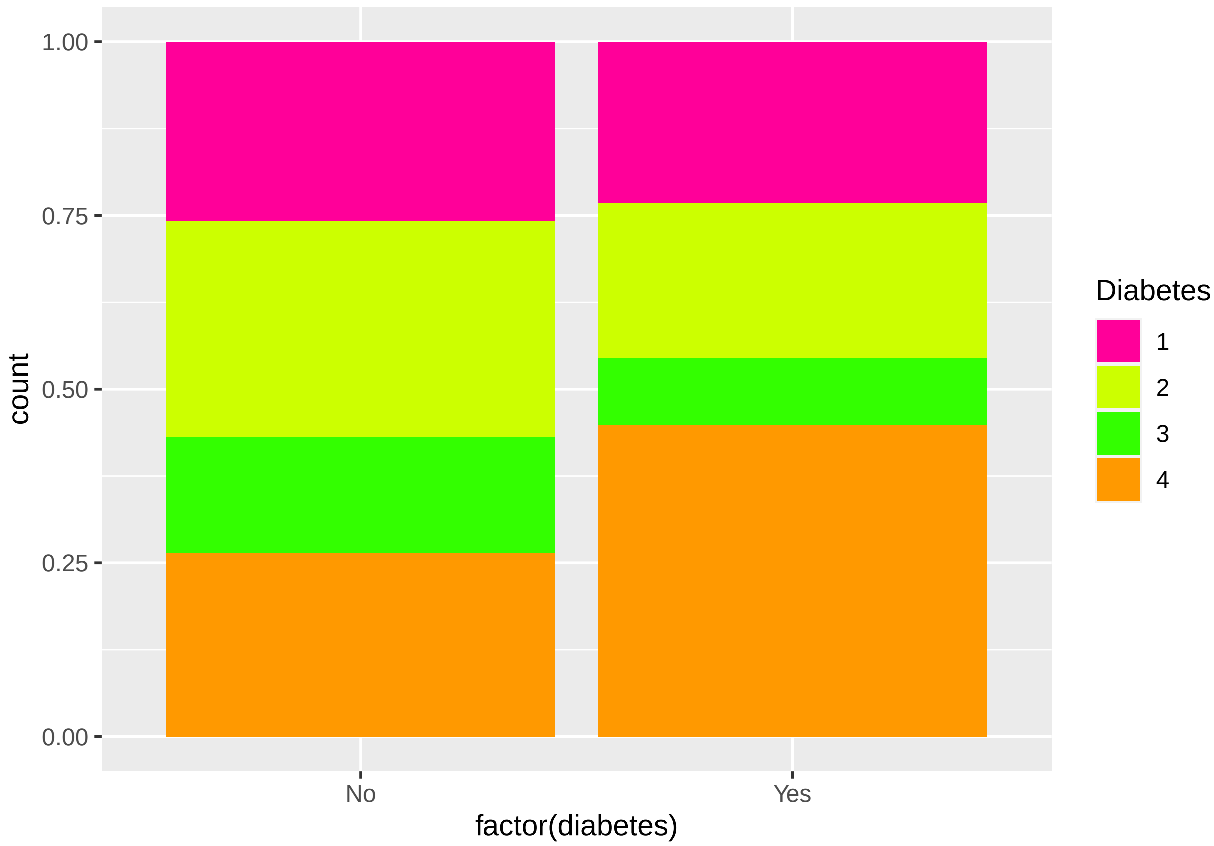
```
aggregate(data_with_clus_assgn[, c('creatinine_phosphokinase')],
list(data_with_clus_assgn$fit.cluster), mean)
```

```
##   Group.1         x
## 1       1 1096.3649
## 2       2  402.1463
## 3       3  408.1707
## 4       4  422.8235
```

We see the high creatinine levels in cluster 1

Diabetes

```
ggplot(data_with_clus_assgn, aes(x = factor(diabetes), fill = fit.cluster))+
  geom_bar(position = "fill")+
  scale_x_discrete(labels  = c("No","Yes"))+
  scale_fill_manual(values = c("#FF0099", "#CCFF00","#33FF00","#FF9900"), name = "Diabetes")
```



We see high diabetic concentration in Cluster 4

ejection_fraction

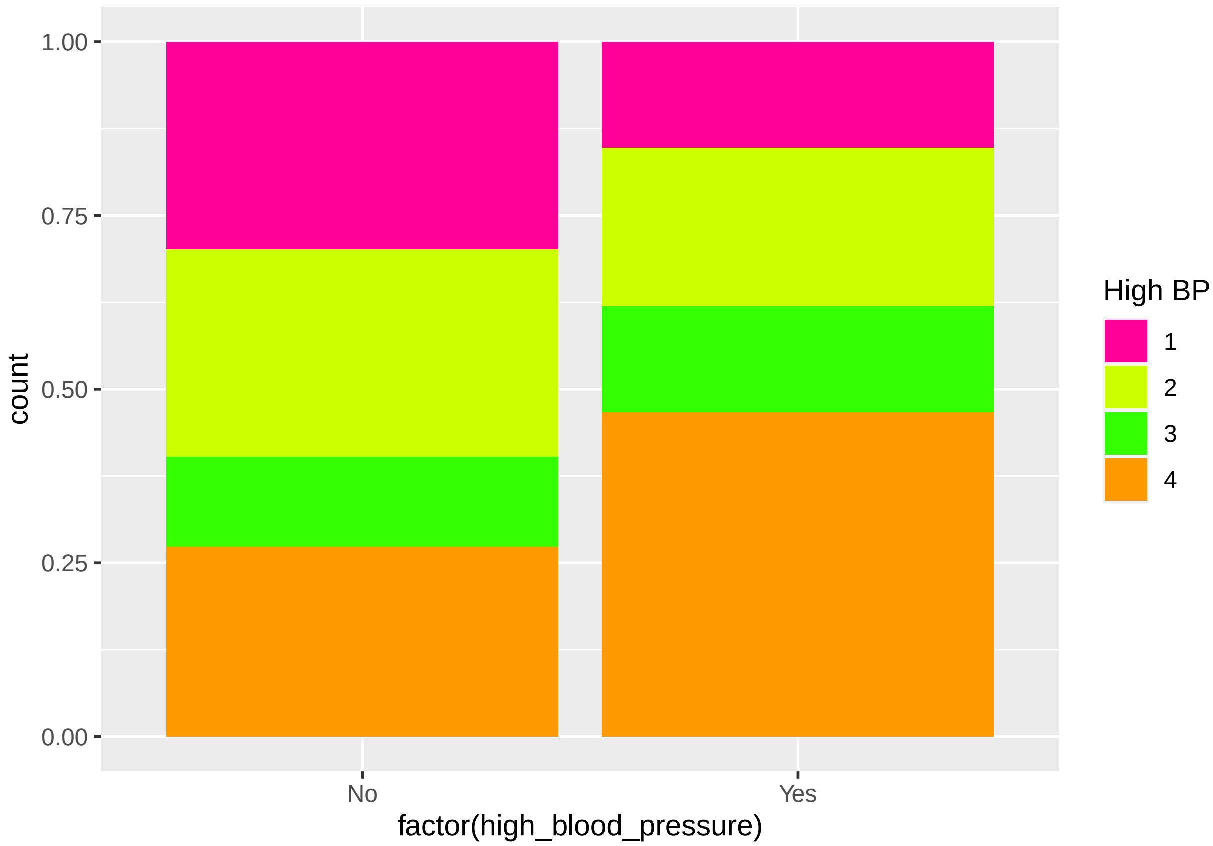
```
aggregate(data_with_clus_assgn[, c('ejection_fraction')],
  list(data_with_clus_assgn$fit.cluster), mean)
```

```
##   Group.1        x
## 1       1 34.55405
## 2       2 37.08537
## 3       3 37.09756
## 4       4 41.84314
```

We see marginally high ejection fraction in cluster 4

High_blood_pressure

```
ggplot(data_with_clus_assgn, aes(x = factor(high_blood_pressure), fill = fit.cluster))+
  geom_bar(position = "fill")+
  scale_x_discrete(labels  = c("No","Yes"))+
  scale_fill_manual(values = c("#FF0099", "#CCFF00","#33FF00","#FF9900"), name = "High BP")
```



We see high BP differ in cluster 4 and also less concentration in Cluster 1

platelets

```
aggregate(data_with_clus_assgn[, c('platelets')],
          list(data_with_clus_assgn$fit.cluster), mean)
```

```
##   Group.1        x
## 1       1 240849.9
## 2       2 266514.9
## 3       3 240221.3
## 4       4 286449.6
```

Not much difference across clusters

serum_creatinine

```
aggregate(data_with_clus_assgn[, c('serum_creatinine')],
          list(data_with_clus_assgn$fit.cluster), mean)
```

```
##   Group.1        x
## 1       1 1.222973
## 2       2 1.187561
## 3       3 2.775366
## 4       4 1.128431
```

We notice high serum_creatinine levels in cluster 3

serum_sodium

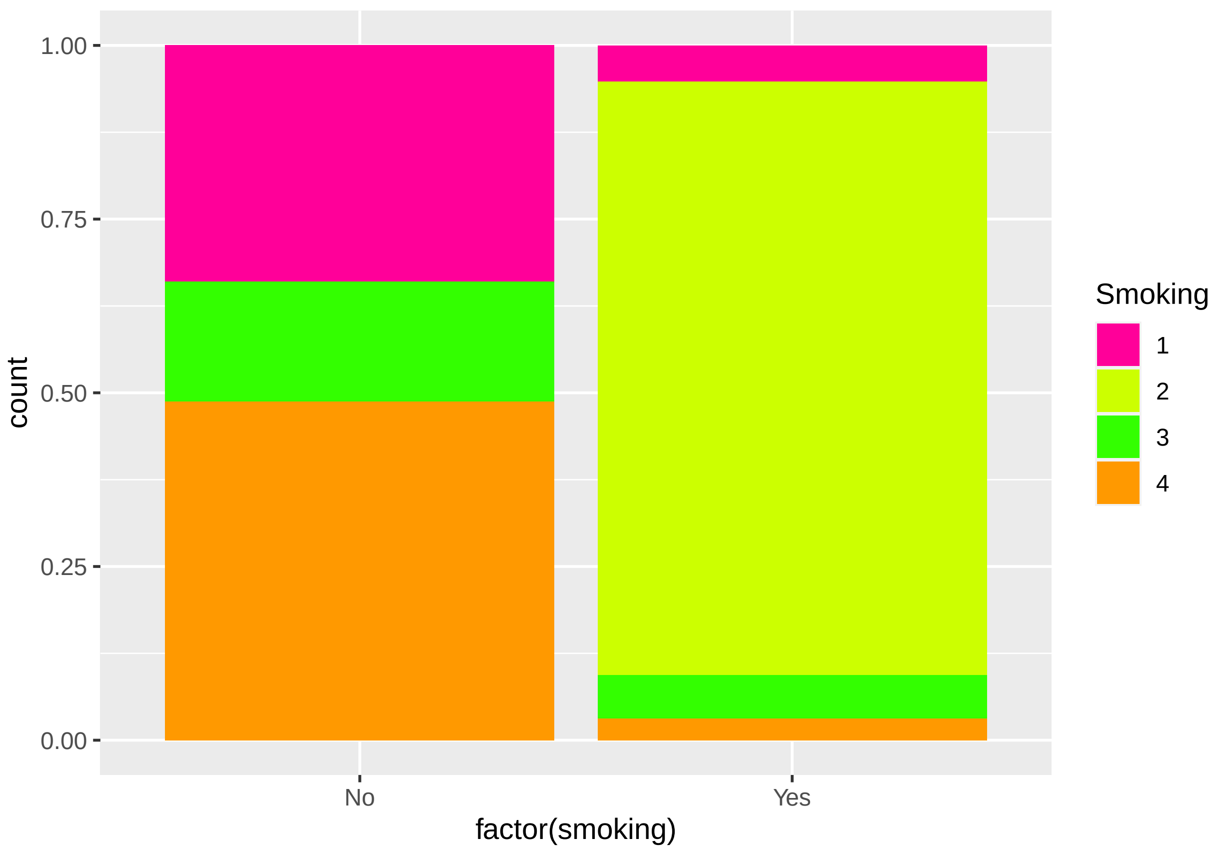
```
aggregate(data_with_clus_assgn[, c('serum_sodium')],
          list(data_with_clus_assgn$fit.cluster), mean)
```

```
##   Group.1        x
## 1       1 136.9865
## 2       2 136.8659
## 3       3 133.3171
## 4       4 137.5000
```

We notice no significant difference across clusters

Smoking

```
ggplot(data_with_clus_assgn, aes(x = factor(smoking), fill = fit.cluster))+
  geom_bar(position = "fill")+
  scale_x_discrete(labels  = c("No","Yes"))+
  scale_fill_manual(values = c("#FF0099", "#CCFF00","#33FF00","#FF9900"),
    name = "Smoking")
```



We notice smoking dominates cluster 2 and is least for cluster 4 and to some extent low for cluster 1 as well

follow-up period

```r
aggregate(data_with_clus_assgn[, c('time')],
          list(data_with_clus_assgn$fit.cluster), mean)
```

```
##   Group.1         x
## 1       1 157.20270
## 2       2 136.35366
## 3       3  60.46341
## 4       4 133.87255
```

We notice high follow up period for cluster 1 and least for cluster 3

Final Profiling Results

Cluster 1 characteristics -

Cluster 1 has high avg. creatinine phosphokinase and a high follow up period with the least death rate and ejection fraction. They are also males with low anaemiac condition and least in age compared to other clusters and have a low % of high bp cases

Single line summary ->

Males with low anaemic issues and low high bp cases with high creatinine phosphokinase and low ejection fraction and have shorter follow up periods

Cluster 2 characteristics -

Cluster 2 has again low anaemic only male population. They also consist only of smokers

Single line summary ->

Only Males who are smokers with low anaemic issues

Cluster 3 characteristics -

Cluster 3 is higher age group, more anaemic, high bp individuals with high serum_creatinine and a low follow up period with the highest death rate

Single line summary ->

Higher age group male dominated with high bp issues, high serum_creatinine and a low follow up period

Cluster 4 characteristics -

Cluster 4 is more diabetic female dominated population with a high ejection_fraction but least smokers

Single line summary ->

Female dominated diabetic individuals with high bp issues and high serum_creatinine and a low follow up period

When we relate these profiles to death events, we see why cluster 3 has disproportionate death event rate as compared to other clusters

This concludes our analysis of Clustering methods for our dataset.