R Notebook

Setting working Directory to Data lOcation

```
library(data.table)
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at
https://goo.gl/13EFCZ
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------
- tidyverse 1.2.1 --
```

```
## v tibble  1.4.1     v purrr   0.2.4
## v tidyr   0.7.2     v stringr 1.2.0
## v readr   1.1.1     v forcats 0.2.0
```

```
## -- Conflicts ------------------------------------------------------------
tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
train = fread("E:/USA/Projects/Research/R_code/w6/train_clust.csv",data.table
= T)
train = train[,-1]
test = fread("E:/USA/Projects/Research/R_code/w6/test_clust.csv",data.table =
```

```
T)
test = test[,-1]
```

Here I have built Custom Function to create Features

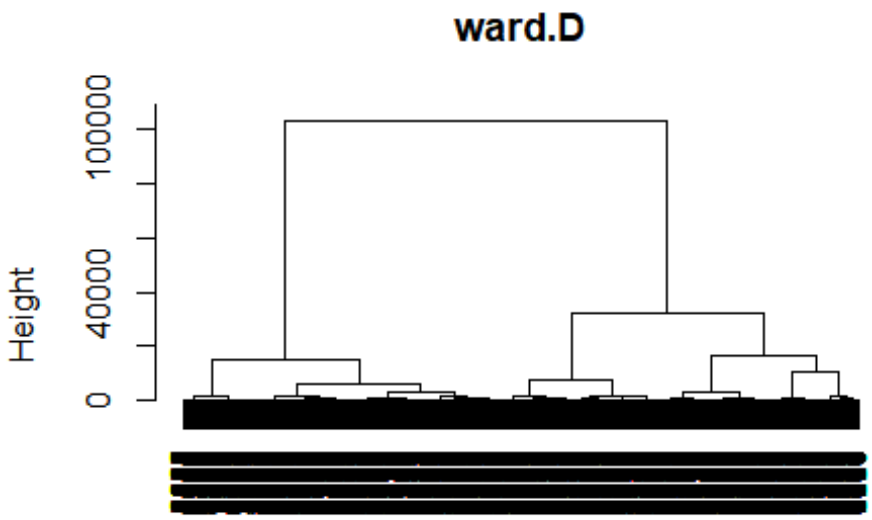
```
features = function(data){
    newdata = NULL
    mean_speed = as.data.frame( rep(0,dim(data)[1]))
    mean_acc_lot =as.data.frame( rep(0,dim(data)[1]))
    mean_acc_lan = as.data.frame(rep(0,dim(data)[1]))
    sd_speed = as.data.frame(rep(0,dim(data)[1]))
    sd_acc_lot = as.data.frame(rep(0,dim(data)[1]))
    sd_acc_lat = as.data.frame(rep(0,dim(data)[1]))
    max_speed = as.data.frame(rep(0,dim(data)[1]))
    max_acc_lot = as.data.frame(rep(0,dim(data)[1]))
    max_acc_lat = as.data.frame(rep(0,dim(data)[1]))
    min_speed = as.data.frame(rep(0,dim(data)[1]))
    min_acc_lot = as.data.frame(rep(0,dim(data)[1]))
    min_acc_lat = as.data.frame(rep(0,dim(data)[1]))
    for (i in c(1:dim(data)[1])) {
        mean_speed[i,] = mean(unlist(data[i,4:64]))
        mean_acc_lot[i,] = mean(unlist(data[i , 65:125]))
        mean_acc_lan[i,] = mean(unlist(data[i, 126:186]))
        sd_speed[i,] = sd((unlist(data[ i,4:64])))
        sd_acc_lot[i,] = sd((unlist(data[i , 65:125])))
        sd_acc_lat[i,] = sd((unlist(data[i , 126:186])))
        max_speed[i,] = max((unlist(data[ i,4:64])))
        max_acc_lot[i,] = max((unlist(data[i , 65:125])))
        max_acc_lat[i,] = max((unlist(data[i , 126:186])))
        min_speed[i,] = min((unlist(data[ i,4:64])))
        min_acc_lot[i,] = min((unlist(data[i , 65:125])))
        min_acc_lat[i,] = min((unlist(data[i , 126:186])))
    }
    newdata =as.data.table(cbind(mean_speed,mean_acc_lot,mean_acc_lan,
sd_speed,sd_acc_lot,sd_acc_lat,

max_speed,max_acc_lot,max_acc_lat,min_speed,mean_acc_lot,mean_acc_lan))
    colnames(newdata) = c("mean_speed","mean_acc_lot","mean_acc_lan",
"sd_speed","sd_acc_lot","sd_acc_lat","max_speed",
                          "max_acc_lot","max_acc_lat","min_speed",
"mean_acc_lot","mean_acc_lan")
    return(newdata)
}
```

Creating Data

```
train_feat = features(train)
test_feat = features(test)

hc_ward=hclust(dist(train_feat), method="ward.D")
```

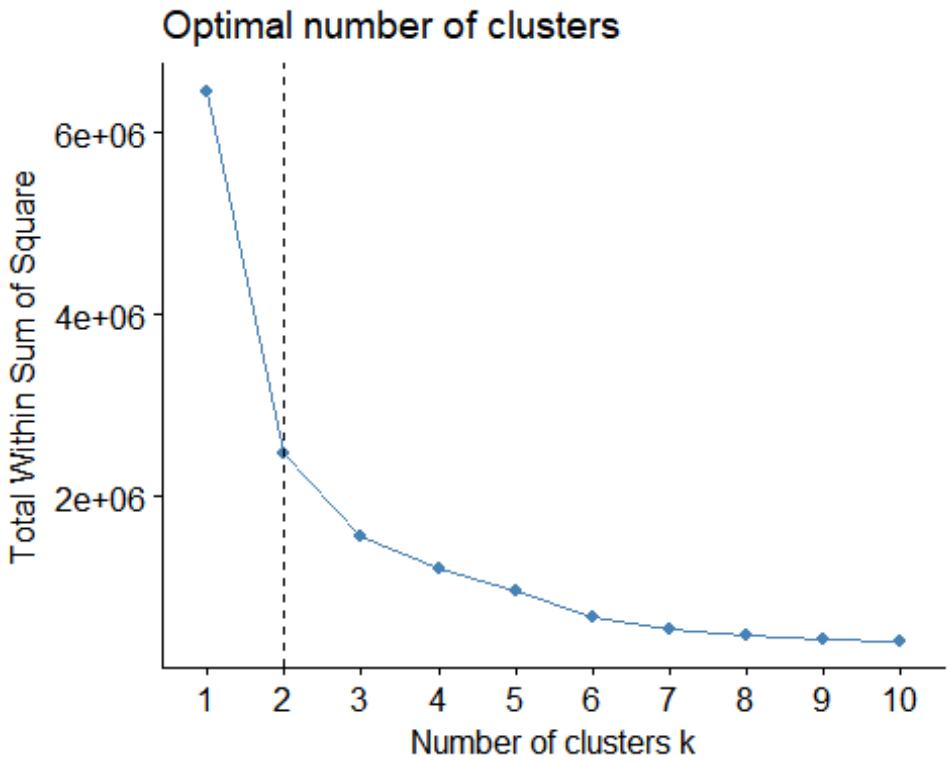
Questions related to type of Dissimilarity measure to use?

```
plot(hc_ward,main="ward.D", xlab="", sub="", cex=.9)
```

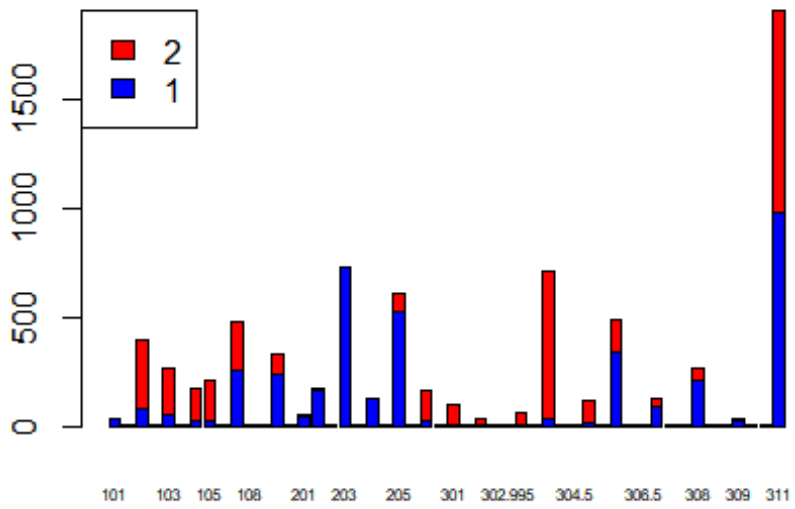


Here we can see only 2 clusters.

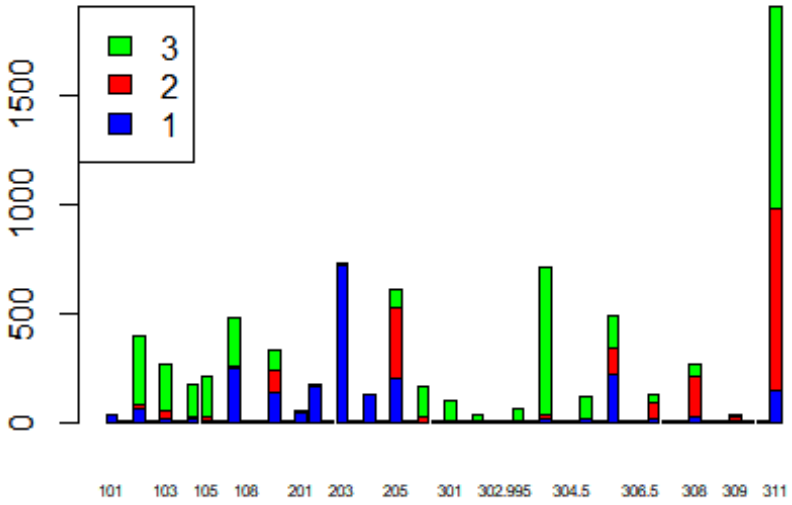
```
fviz_nbclust(train_feat, hcut, method = "wss",hc_method = "ward.D", main =
"Ward.D") +
  geom_vline(xintercept = 2, linetype = 2)
```




```
hc_ward_cut = cutree(hc_ward,k=2)
hc_ward_cut2 = cutree(hc_ward,k=3)

index = fread("E:/USA/Projects/Research/R_code/w6/index_all.csv")
index$new = paste(index$RunName,index$win60s)
train$new = paste(train$RunName,train$win60s)

index_train = as.data.table(train$new)
colnames(index_train)='new'
index_train = merge(index_train, index[,c('c','new')],by = 'new')

index_train1 = cbind(index_train,as.data.table(hc_ward_cut))

index_train1$c = as.factor(index_train1$c)
index_train1$hc_ward_cut =as.factor(index_train1$hc_ward_cut)

ind = group_by(index_train1[,2:3], c) %>% summarize(size =
length(hc_ward_cut), frq1 = summary(hc_ward_cut)[1],frq2 =
summary(hc_ward_cut)[2])

barplot(height = t(ind[,c(3,4)]), names.arg =
ind$c,col=c("blue","red"),legend.text = c("1","2"),args.legend = list(x =
"topleft"),axisnames = T,cex.names = 0.5)
```




```
index_train2 = cbind(index_train,as.data.table(hc_ward_cut2))
index_train2$c = as.factor(index_train2$c)
index_train2$hc_ward_cut2 =as.factor(index_train2$hc_ward_cut2)

ind1 = group_by(index_train2[,2:3], c) %>% summarize(size =
length(hc_ward_cut2), frq1 = summary(hc_ward_cut2)[1],frq2 =
summary(hc_ward_cut2)[2],frq3 = summary(hc_ward_cut2)[3])

barplot(height = t(ind1[,c(3,4,5)]), names.arg =
ind1$c,col=c("blue","red","green"),legend.text = c("1","2","3"),args.legend =
list(x = "topleft"),axisnames = T,cex.names = 0.5)
```


3
2
1
0
500
1000
1500
101
103
105
108
201
203
205
301
302.995
304.5
306.5
308
309
311