Clustering of Events

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/w8/train\_clust.csv",data.table = T)  
train = train[,-1]  
test = fread("E:/USA/Projects/Research/R\_code/w8/test\_clust.csv",data.table = T)  
test = test[,-1]

#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')

event\_count = as.data.table(table(train$Event\_ID, dnn = c("events")))  
as.data.table(table(train$Event\_ID, dnn = c("events")))

## events N  
## 1: 101 34  
## 2: 102 408  
## 3: 103 264  
## 4: 104 172  
## 5: 105 197  
## 6: 106 501  
## 7: 111 341  
## 8: 202 191  
## 9: 203 736  
## 10: 204 130  
## 11: 205 608  
## 12: 206 170  
## 13: 301 128  
## 14: 303 22  
## 15: 304 748  
## 16: 305 114  
## 17: 306 501  
## 18: 307 108  
## 19: 308 285  
## 20: 311 1926

events\_more\_than\_10 = 25 < event\_count$N & event\_count$N <= 100  
events\_more\_than\_10 = event\_count$events[events\_more\_than\_10]  
  
events\_more\_than\_100 = event\_count$N > 100  
events\_more\_than\_100 = event\_count$events[events\_more\_than\_100]

#this is for events having length greater than 100  
sample\_index = NULL  
for (i in 1:length(events\_more\_than\_100)){  
 set.seed(1001)  
 temp = filter(train, Event\_ID == events\_more\_than\_100[i]) %>% sample\_n(100)  
 sample\_index = rbind(sample\_index, temp)  
}  
  
#this is for variable length less than 100  
sampl\_ind = NULL  
for (i in 1:length(events\_more\_than\_10)) {  
 temp = filter(train, Event\_ID == events\_more\_than\_10[i])  
 sampl\_ind = rbind(sampl\_ind, temp)  
  
}  
#total sample index for clustering   
join\_train = rbind(sampl\_ind, sample\_index)

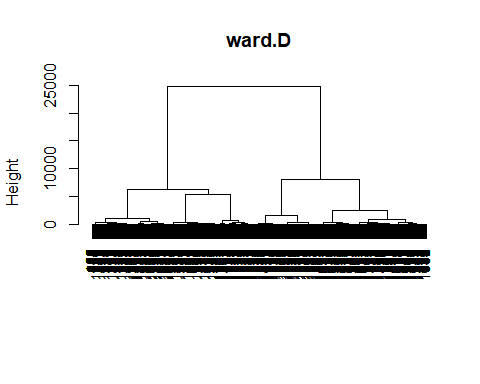
features = function(data){  
 newdata = NULL  
 mean\_speed = as.data.frame( rep(0,dim(data)[1]))  
 mean\_acc\_lot =as.data.frame( rep(0,dim(data)[1]))  
 mean\_acc\_lan = as.data.frame(rep(0,dim(data)[1]))  
 sd\_speed = as.data.frame(rep(0,dim(data)[1]))  
 sd\_acc\_lot = as.data.frame(rep(0,dim(data)[1]))  
 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,5:65]))  
 mean\_acc\_lot[i,] = mean(unlist(data[i , 66:126]))  
 mean\_acc\_lan[i,] = mean(unlist(data[i, 127:187]))  
 sd\_speed[i,] = sd((unlist(data[ i,5:65])))  
 sd\_acc\_lot[i,] = sd((unlist(data[i , 66:126])))  
 sd\_acc\_lat[i,] = sd((unlist(data[i , 127:187])))  
 max\_speed[i,] = max((unlist(data[ i,5:65])))  
 max\_acc\_lot[i,] = max((unlist(data[i , 66:126])))  
 max\_acc\_lat[i,] = max((unlist(data[i , 127:187])))  
 min\_speed[i,] = min((unlist(data[ i,5:65])))  
 min\_acc\_lot[i,] = min((unlist(data[i , 66:126])))  
 min\_acc\_lat[i,] = min((unlist(data[i , 127:187])))  
 }  
 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)  
}

train\_feat = features(join\_train)

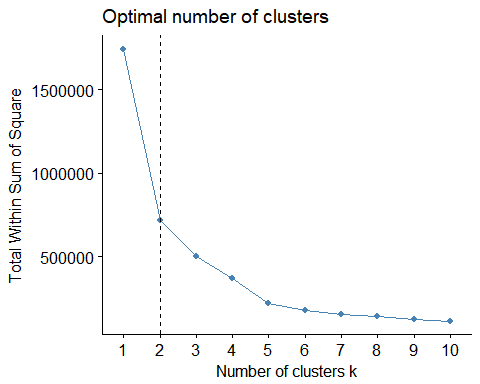
hc\_ward=hclust(dist(train\_feat), method="ward.D")

### Questions related to type of Dissimilarity measure to use?

plot(hc\_ward,main="ward.D", xlab="", sub="", cex=.9)

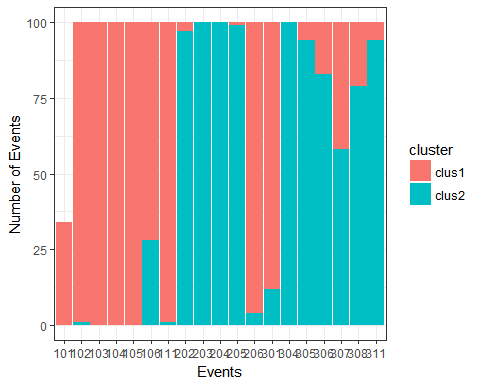
 Here we can see only 2 clusters.

fviz\_nbclust(train\_feat, hcut, method = "wss",hc\_method = "ward.D", main = "Ward.D") +  
 geom\_vline(xintercept = 2, linetype = 2)

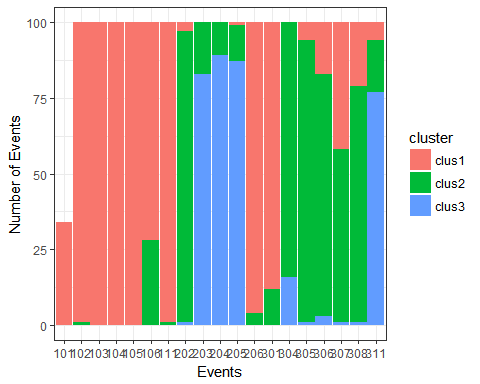


hc\_ward\_cut2 = cutree(hc\_ward,k = 2)  
hc\_ward\_cut3 = cutree(hc\_ward,k = 3)  
hc\_ward\_cut4 = cutree(hc\_ward,k = 4)

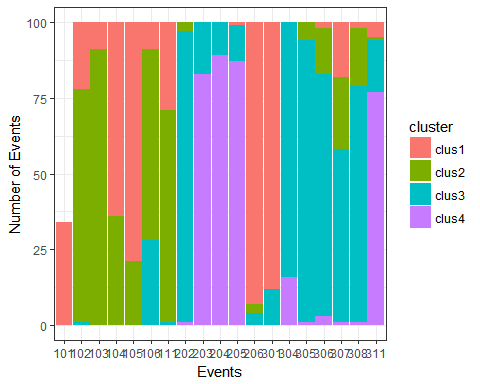
index\_train2 = cbind(join\_train[,'Event\_ID'],as.data.table(hc\_ward\_cut2))  
index\_train2$V1 = as.factor(index\_train2$V1)  
index\_train2$hc\_ward\_cut2 =as.factor(index\_train2$hc\_ward\_cut2)  
  
ind1 = group\_by(index\_train2, V1) %>% summarize(size = length(hc\_ward\_cut2), clus1 = summary(hc\_ward\_cut2)[1],clus2 = summary(hc\_ward\_cut2)[2])  
  
ind\_new1 = ind1 %>% gather(`clus1`, `clus2`, key = cluster, value = count)  
ind\_new1$size = NULL  
ind\_new1$V1 = as.factor(ind\_new1$V1)  
ggplot(ind\_new1, aes(x=V1, y=count, fill = cluster, label = "cluster1","cluster2")) + geom\_bar(stat="identity",position="stack",width=0.95)+theme\_bw() + ylab("Number of Events") +xlab("Events")



index\_train3 = cbind(join\_train[,'Event\_ID'],as.data.table(hc\_ward\_cut3))  
index\_train3$V1 = as.factor(index\_train3$V1)  
index\_train3$hc\_ward\_cut3 =as.factor(index\_train3$hc\_ward\_cut3)  
  
ind2 = group\_by(index\_train3, V1) %>% summarize(size = length(hc\_ward\_cut3), clus1 = summary(hc\_ward\_cut3)[1],clus2 = summary(hc\_ward\_cut3)[2],clus3 = summary(hc\_ward\_cut3)[3])  
  
ind2 = ind2 %>% gather(`clus1`, `clus2`, `clus3`, key = cluster, value = count)  
ind2$size = NULL  
ind2$V1 = as.factor(ind2$V1)  
ggplot(ind2, aes(x=V1, y=count, fill = cluster, label = "cluster1","cluster2")) + geom\_bar(stat="identity",position="stack",width=0.95)+theme\_bw() + ylab("Number of Events") +xlab("Events")



index\_train4 = cbind(join\_train[,'Event\_ID'],as.data.table(hc\_ward\_cut4))  
index\_train4$V1 = as.factor(index\_train4$V1)  
index\_train4$hc\_ward\_cut4 =as.factor(index\_train4$hc\_ward\_cut4)  
  
ind3 = group\_by(index\_train4, V1) %>% summarize(clus1 = sum(hc\_ward\_cut4==1),clus2 = sum(hc\_ward\_cut4==2),clus3 = sum(hc\_ward\_cut4==3), clus4 = sum(hc\_ward\_cut4==4), clus5 = sum(hc\_ward\_cut4==5))  
  
  
ind3 = ind3 %>% gather(`clus1`, `clus2`, `clus3`,`clus4`, key = cluster, value = count)  
ind3$size = NULL  
ind3$V1 = as.factor(ind3$V1)  
ggplot(ind3, aes(x=V1, y=count, fill = cluster, label = "cluster1","cluster2")) + geom\_bar(stat="identity",position="stack",width=0.95)+theme\_bw() + ylab("Number of Events") +xlab("Events")



indices = ifelse(hc\_ward\_cut2==1,TRUE, FALSE)  
summary(as.factor(hc\_ward\_cut2))

## 1 2   
## 884 950

summary(indices)

## Mode FALSE TRUE NA's   
## logical 950 884 0

clust\_1 = train\_feat[indices]  
clust\_2 = train\_feat[!indices]

print("Results before Clusters, mean and Std Dev")

## [1] "Results before Clusters, mean and Std Dev"

mean(train\_feat$mean\_speed)

## [1] 39.4414

sd(train\_feat$mean\_speed)

## [1] 15.86829

print("~~~~~~")

## [1] "~~~~~~"

print("mean speed of cluster 1")

## [1] "mean speed of cluster 1"

mean(clust\_1$mean\_speed)

## [1] 26.33566

print("mean speed of Cluster 2")

## [1] "mean speed of Cluster 2"

mean(clust\_2$mean\_speed)

## [1] 51.63663

print("~~~~~~~~~")

## [1] "~~~~~~~~~"

print("Std Dev speed of cluster 1")

## [1] "Std Dev speed of cluster 1"

sd(clust\_1$mean\_speed)

## [1] 8.670452

print("Std Dev of Cluster 2")

## [1] "Std Dev of Cluster 2"

sd(clust\_2$mean\_speed)

## [1] 10.36985

mean(clust\_1$sd\_speed)

## [1] 7.632576

mean(clust\_2$sd\_speed)

## [1] 3.350743

indices1 = ifelse(hc\_ward\_cut3==1,TRUE, FALSE)  
indices2 = ifelse(hc\_ward\_cut3==2,TRUE, FALSE)  
indices3 = ifelse(hc\_ward\_cut3==3,TRUE, FALSE)  
summary(as.factor(hc\_ward\_cut3))

## 1 2 3   
## 884 591 359

clust\_1 = train\_feat[indices1]  
clust\_2 = train\_feat[indices2]  
clust\_3 = train\_feat[indices3]

print("Results before Clusters, mean and Std Dev")

## [1] "Results before Clusters, mean and Std Dev"

mean(train\_feat$mean\_speed)

## [1] 39.4414

sd(train\_feat$mean\_speed)

## [1] 15.86829

print("~~~~~~")

## [1] "~~~~~~"

print("mean speed of cluster 1")

## [1] "mean speed of cluster 1"

mean(clust\_1$mean\_speed)

## [1] 26.33566

print("mean speed of Cluster 2")

## [1] "mean speed of Cluster 2"

mean(clust\_2$mean\_speed)

## [1] 45.12913

print("mean speed of Cluster 3")

## [1] "mean speed of Cluster 3"

mean(clust\_3$mean\_speed)

## [1] 62.34955

print("~~~~~~~~~")

## [1] "~~~~~~~~~"

print("Std Dev speed of cluster 1")

## [1] "Std Dev speed of cluster 1"

sd(clust\_1$mean\_speed)

## [1] 8.670452

print("Std Dev of Cluster 2")

## [1] "Std Dev of Cluster 2"

sd(clust\_2$mean\_speed)

## [1] 6.477981

print("Std Dev of Cluster 3")

## [1] "Std Dev of Cluster 3"

sd(clust\_3$mean\_speed)

## [1] 5.558739

mean(clust\_1$sd\_speed)

## [1] 7.632576

mean(clust\_2$sd\_speed)

## [1] 4.187965

mean(clust\_3$sd\_speed)

## [1] 1.972474

indices1 = ifelse(hc\_ward\_cut4==1,TRUE, FALSE)  
indices2 = ifelse(hc\_ward\_cut4==2,TRUE, FALSE)  
indices3 = ifelse(hc\_ward\_cut4==3,TRUE, FALSE)  
indices4 = ifelse(hc\_ward\_cut4==4,TRUE, FALSE)  
summary(as.factor(hc\_ward\_cut4))

## 1 2 3 4   
## 455 429 591 359

clust\_1 = train\_feat[indices1]  
clust\_2 = train\_feat[indices2]  
clust\_3 = train\_feat[indices3]  
clust\_4 = train\_feat[indices4]

print("Results before Clusters, mean and Std Dev")

## [1] "Results before Clusters, mean and Std Dev"

mean(train\_feat$mean\_speed)

## [1] 39.4414

sd(train\_feat$mean\_speed)

## [1] 15.86829

print("~~~~~~")

## [1] "~~~~~~"

print("mean speed of cluster 1")

## [1] "mean speed of cluster 1"

mean(clust\_1$mean\_speed)

## [1] 26.12025

print("mean speed of Cluster 2")

## [1] "mean speed of Cluster 2"

mean(clust\_2$mean\_speed)

## [1] 26.56413

print("mean speed of Cluster 3")

## [1] "mean speed of Cluster 3"

mean(clust\_3$mean\_speed)

## [1] 45.12913

print("mean speed of Cluster 4")

## [1] "mean speed of Cluster 4"

mean(clust\_4$mean\_speed)

## [1] 62.34955

print("~~~~~~~~~")

## [1] "~~~~~~~~~"

print("Std Dev speed of cluster 1")

## [1] "Std Dev speed of cluster 1"

sd(clust\_1$mean\_speed)

## [1] 11.53161

print("Std Dev of Cluster 2")

## [1] "Std Dev of Cluster 2"

sd(clust\_2$mean\_speed)

## [1] 3.733401

print("Std Dev of Cluster 3")

## [1] "Std Dev of Cluster 3"

sd(clust\_3$mean\_speed)

## [1] 6.477981

print("Std Dev of Cluster 4")

## [1] "Std Dev of Cluster 4"

sd(clust\_4$mean\_speed)

## [1] 5.558739

mean(clust\_1$sd\_speed)

## [1] 12.70411

mean(clust\_2$sd\_speed)

## [1] 2.253674

mean(clust\_3$sd\_speed)

## [1] 4.187965

mean(clust\_4$sd\_speed)

## [1] 1.972474