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/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]

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(index\_train$c, dnn = c("events")))  
as.data.table(table(index\_train$c, dnn = c("events")))

## events N  
## 1: 101 31  
## 2: 101.5 1  
## 3: 102 397  
## 4: 102.5 3  
## 5: 103 267  
## 6: 103.5 4  
## 7: 104 171  
## 8: 105 205  
## 9: 105.5 1  
## 10: 106 475  
## 11: 108 2  
## 12: 110.99 1  
## 13: 111 331  
## 14: 200.995 1  
## 15: 201 47  
## 16: 202 169  
## 17: 202.5 2  
## 18: 203 732  
## 19: 203.5 6  
## 20: 204 129  
## 21: 204.5 1  
## 22: 205 604  
## 23: 205.5 2  
## 24: 206 160  
## 25: 253.5 2  
## 26: 301 95  
## 27: 301.995 1  
## 28: 302 31  
## 29: 302.5 3  
## 30: 302.995 1  
## 31: 303 56  
## 32: 303.5 2  
## 33: 304 712  
## 34: 304.01 1  
## 35: 304.5 2  
## 36: 305 118  
## 37: 305.5 3  
## 38: 306 484  
## 39: 306.01 1  
## 40: 306.5 1  
## 41: 307 127  
## 42: 307.03 1  
## 43: 307.5 1  
## 44: 308 262  
## 45: 308.01 1  
## 46: 308.49 1  
## 47: 309 31  
## 48: 309.005 1  
## 49: 310 1  
## 50: 311 1903  
## events N

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 = c()  
for (i in 1:length(events\_more\_than\_100)){  
 set.seed(1001)  
 s\_index = sample(index\_train$new[index\_train$c == events\_more\_than\_100[i]],100)  
 sample\_index = c(sample\_index, s\_index)  
   
}  
  
#this is for variable length less than 100  
sampl\_ind = c()  
for (i in 1:length(events\_more\_than\_10)) {  
 s\_index = index\_train$new[index\_train$c == events\_more\_than\_10[i]]  
 sampl\_ind = c(sampl\_ind, s\_index)  
  
  
#total sample index for clustering   
total\_ind = c(sampl\_ind, sample\_index)   
}

temp = as.data.frame(total\_ind,col.names = "new")  
colnames(temp) = "new"  
join\_temp = semi\_join(index\_train, temp, by = "new")

## Warning: Column `new` joining character vector and factor, coercing into  
## character vector

as.data.frame(table(join\_temp$c))

## Var1 Freq  
## 1 101 31  
## 2 102 100  
## 3 103 100  
## 4 104 100  
## 5 105 100  
## 6 106 100  
## 7 111 100  
## 8 201 47  
## 9 202 100  
## 10 203 100  
## 11 204 100  
## 12 205 100  
## 13 206 100  
## 14 301 95  
## 15 302 31  
## 16 303 56  
## 17 304 100  
## 18 305 100  
## 19 306 100  
## 20 307 100  
## 21 308 100  
## 22 309 31  
## 23 311 100

join\_train = semi\_join(train, temp, by = "new")

## Warning: Column `new` joining character vector and factor, coercing into  
## character vector

join\_train$new = NULL  
  
events\_train = inner\_join(temp, index\_train, by = "new")

## Warning: Column `new` joining factor and character vector, coercing into  
## character vector

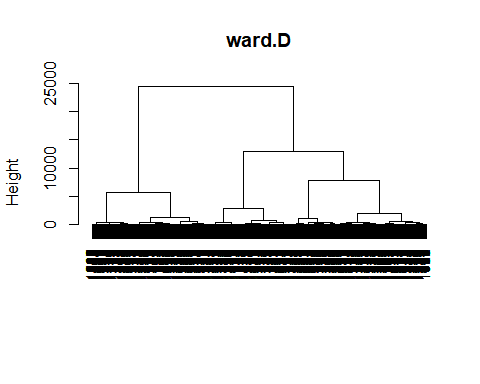
shannon.entropy <- function(p)  
{  
 if (min(p) < 0 || sum(p) <= 0)  
 return(NA)  
 p.norm <- p[p>0]/sum(p)  
 -sum(log2(p.norm)\*p.norm)  
}  
features = function(data){  
 newdata = NULL  
 mean\_speed = as.data.frame( rep(0,dim(data)[1]))  
 mean\_acc\_lot =as.data.frame( rep(0,dim(data)[1]))  
 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]))  
 shenen\_speed = as.data.frame(rep(0,dim(data)[1]))  
 shenen\_acc\_lot = as.data.frame(rep(0,dim(data)[1]))  
 shenen\_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])))  
 shenen\_speed[i,] = shannon.entropy((unlist(data[ i,4:64])))  
 shenen\_acc\_lot[i,] = shannon.entropy((unlist(data[i , 65:125])))  
 shenen\_acc\_lat[i,] = shannon.entropy((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,  
 shenen\_speed,shenen\_acc\_lot,shenen\_acc\_lat))  
 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",  
 "shenen\_speed","shenen\_acc\_lot","shenen\_acc\_lat")  
 return(newdata)  
}

train\_feat = features(join\_train)

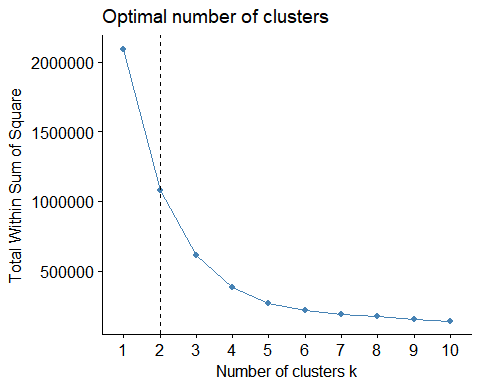
hc\_ward=hclust(dist(train\_feat), method="ward.D")

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

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

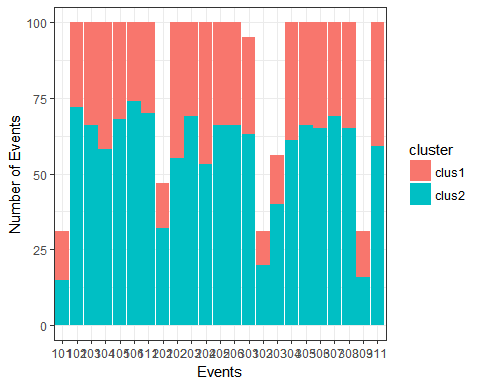
 Here we can see only 2 clusters.

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

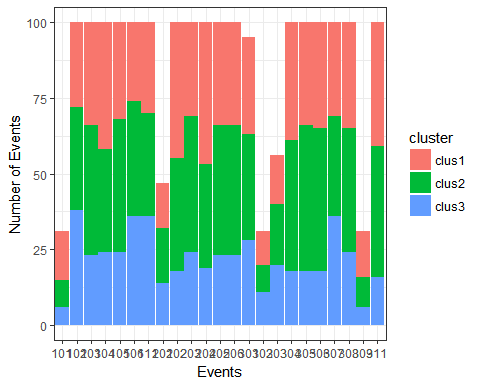


hc\_ward\_cut2 = cutree(hc\_ward,k = 2)  
hc\_ward\_cut3 = cutree(hc\_ward,k = 3)  
hc\_ward\_cut4 = cutree(hc\_ward,k = 4)

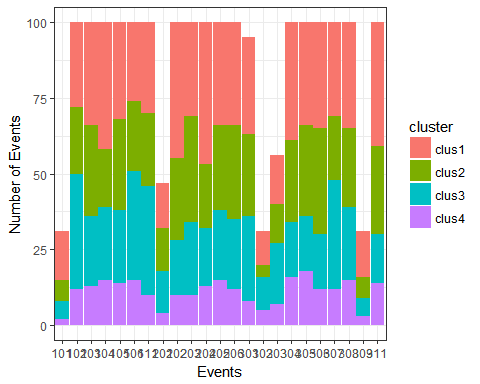
index\_train2 = cbind(events\_train[,2],as.data.table(hc\_ward\_cut2))  
index\_train2$V1 = as.factor(index\_train2$V1)  
index\_train2$hc\_ward\_cut2 =as.factor(index\_train2$hc\_ward\_cut2)  
  
ind1 = group\_by(index\_train2, V1) %>% summarize(size = length(hc\_ward\_cut2), clus1 = summary(hc\_ward\_cut2)[1],clus2 = 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(events\_train[,2],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(events\_train[,2],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   
## 703 1288

summary(indices)

## Mode FALSE TRUE NA's   
## logical 1288 703 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] 39.99497

sd(train\_feat$mean\_speed)

## [1] 15.4165

print("~~~~~~")

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

print("mean speed of cluster 1")

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

mean(clust\_1$mean\_speed)

## [1] 23.65804

print("mean speed of Cluster 2")

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

mean(clust\_2$mean\_speed)

## [1] 48.91179

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

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

print("Std Dev speed of cluster 1")

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

sd(clust\_1$mean\_speed)

## [1] 5.852188

print("Std Dev of Cluster 2")

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

sd(clust\_2$mean\_speed)

## [1] 11.11088

mean(clust\_1$sd\_speed)

## [1] 4.942401

mean(clust\_2$sd\_speed)

## [1] 5.768045

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   
## 703 785 503

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] 39.99497

sd(train\_feat$mean\_speed)

## [1] 15.4165

print("~~~~~~")

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

print("mean speed of cluster 1")

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

mean(clust\_1$mean\_speed)

## [1] 23.65804

print("mean speed of Cluster 2")

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

mean(clust\_2$mean\_speed)

## [1] 41.71187

print("mean speed of Cluster 3")

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

mean(clust\_3$mean\_speed)

## [1] 60.14825

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

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

print("Std Dev speed of cluster 1")

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

sd(clust\_1$mean\_speed)

## [1] 5.852188

print("Std Dev of Cluster 2")

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

sd(clust\_2$mean\_speed)

## [1] 6.793435

print("Std Dev of Cluster 3")

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

sd(clust\_3$mean\_speed)

## [1] 6.070415

mean(clust\_1$sd\_speed)

## [1] 4.942401

mean(clust\_2$sd\_speed)

## [1] 7.806841

mean(clust\_3$sd\_speed)

## [1] 2.586226

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   
## 703 530 503 255

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] 39.99497

sd(train\_feat$mean\_speed)

## [1] 15.4165

print("~~~~~~")

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

print("mean speed of cluster 1")

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

mean(clust\_1$mean\_speed)

## [1] 23.65804

print("mean speed of Cluster 2")

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

mean(clust\_2$mean\_speed)

## [1] 43.67728

print("mean speed of Cluster 3")

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

mean(clust\_3$mean\_speed)

## [1] 60.14825

print("mean speed of Cluster 4")

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

mean(clust\_4$mean\_speed)

## [1] 37.62689

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

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

print("Std Dev speed of cluster 1")

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

sd(clust\_1$mean\_speed)

## [1] 5.852188

print("Std Dev of Cluster 2")

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

sd(clust\_2$mean\_speed)

## [1] 5.137351

print("Std Dev of Cluster 3")

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

sd(clust\_3$mean\_speed)

## [1] 6.070415

print("Std Dev of Cluster 4")

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

sd(clust\_4$mean\_speed)

## [1] 7.916425

mean(clust\_1$sd\_speed)

## [1] 4.942401

mean(clust\_2$sd\_speed)

## [1] 4.023099

mean(clust\_3$sd\_speed)

## [1] 2.586226

mean(clust\_4$sd\_speed)

## [1] 15.67109