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
```

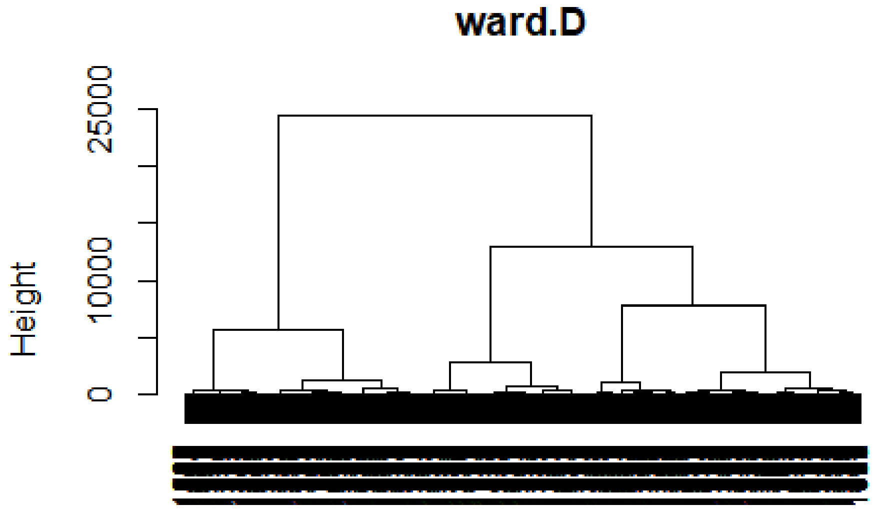
```
join_train$new = NULL

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")
```

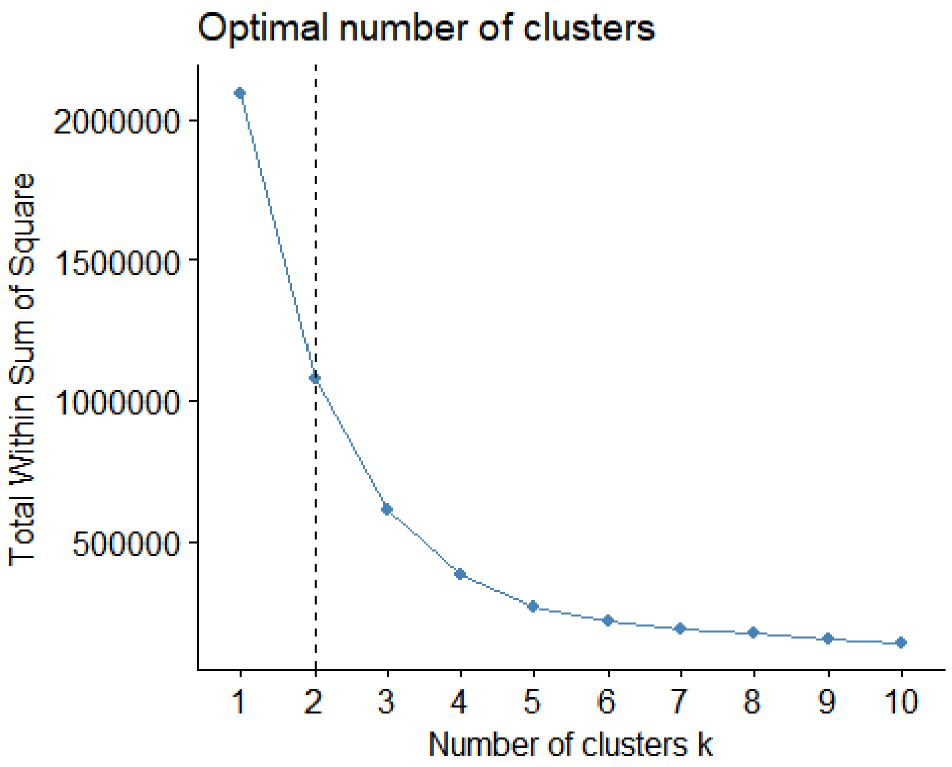
Questions related to type of Dissimilarity measure to use?

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

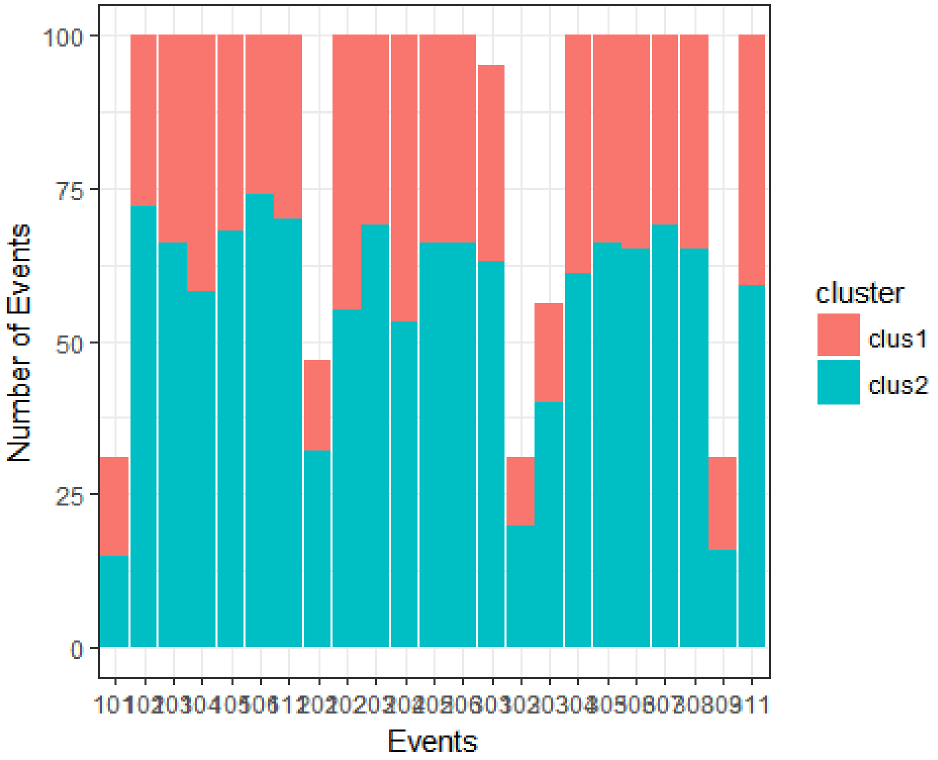


Here we can see only 2 clusters.

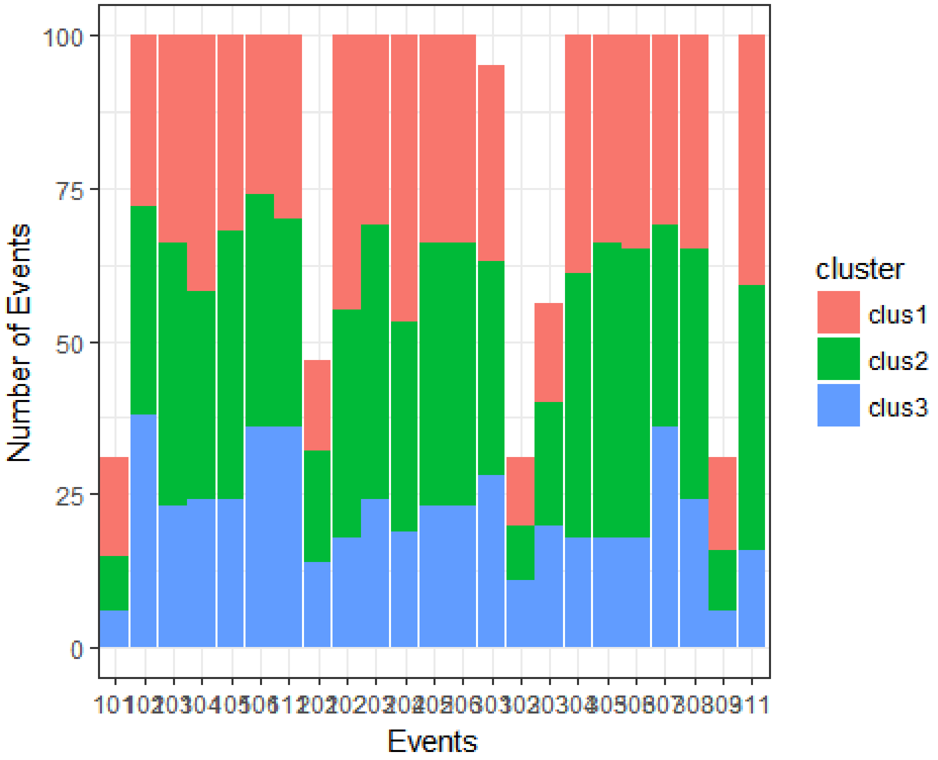
```r
fviz_nbclust(train_feat, hcut, method = "wss",hc_method = "ward.D", main =
"Ward.D") +
  geom_vline(xintercept = 2, linetype = 2)
```



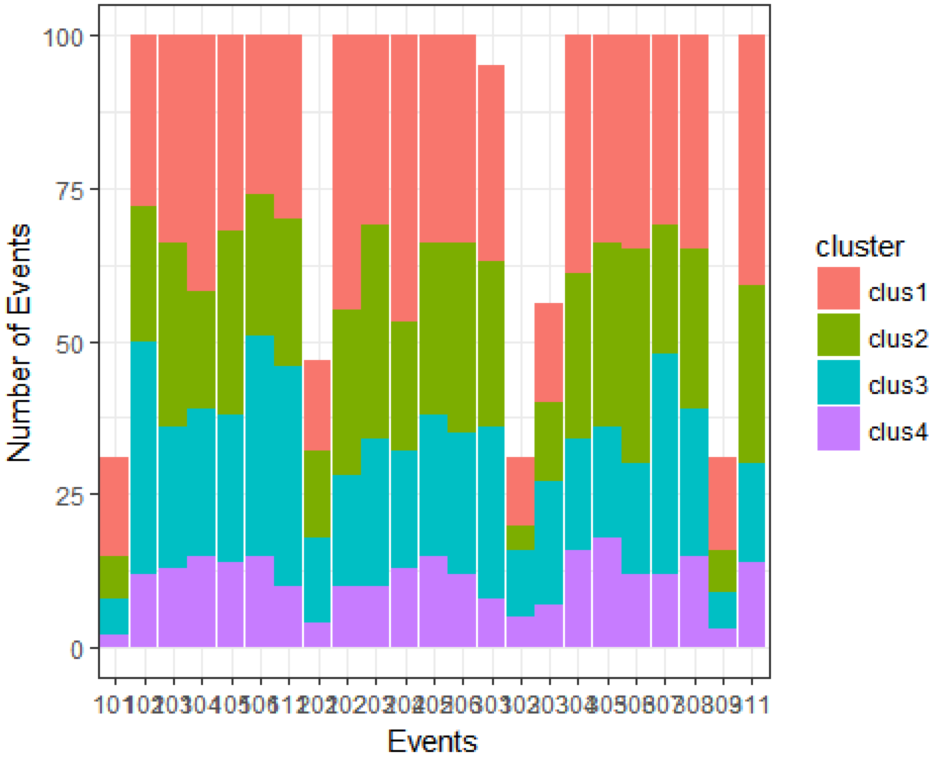
```
hc_ward_cut2 = cutree(hc_ward,k = 2)
hc_ward_cut3 = cutree(hc_ward,k = 3)
hc_ward_cut4 = cutree(hc_ward,k = 4)

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]
```

```r
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
```