pokemon.R

91863

2021-01-31

```
############## dataset information ############
# This dataset contains information on all 800 Pokemon from all six Generations of Pokemon.
  # Is it possible to build a classifier to identify legendary Pokemon?

##Dependent variable: LEGENDARY(yes or no)

## we will do following steps
# 1.import the data set
# 2.remove the unnecessary column
# 3.check the data types,dim,summary
# 4.check the missing values
# 5.check the outliers (replace with mean)
# 6.histograms plots for pokemons
# 7.check the legendary pokemon per generation
# 8.categorical in to numerical featurer
# 9.normalize the data
# 10.apply the kmeans clustering algorithms
# 11.apply the hierarchial clustering algorithms

#read the data
pokemon=read.csv("pokemon.csv")
View(pokemon)
dim(pokemon)
```

```
## [1] 800  13
```

```
#remove unnecessary columns
pokemon=pokemon[-1:-4]

# summary of data
summary(pokemon)
```

```
##      Total             HP              Attack       Defense
##  Min.   :180.0   Min.   :  1.00   Min.   :  5   Min.   :  5.00
##  1st Qu.:330.0   1st Qu.: 50.00   1st Qu.: 55   1st Qu.: 50.00
##  Median :450.0   Median : 65.00   Median : 75   Median : 70.00
##  Mean   :435.1   Mean   : 69.26   Mean   : 79   Mean   : 73.84
##  3rd Qu.:515.0   3rd Qu.: 80.00   3rd Qu.:100   3rd Qu.: 90.00
##  Max.   :780.0   Max.   :255.00   Max.   :190   Max.   :230.00
##     Sp..Atk          Sp..Def          Speed          Generation
##  Min.   : 10.00   Min.   : 20.0   Min.   :  5.00   Min.   :1.000
```

```
##  1st Qu.: 49.75   1st Qu.: 50.0   1st Qu.: 45.00   1st Qu.:2.000
##  Median : 65.00   Median : 70.0   Median : 65.00   Median :3.000
##  Mean   : 72.82   Mean   : 71.9   Mean   : 68.28   Mean   :3.324
##  3rd Qu.: 95.00   3rd Qu.: 90.0   3rd Qu.: 90.00   3rd Qu.:5.000
##  Max.   :194.00   Max.   :230.0   Max.   :180.00   Max.   :6.000
##   Legendary
##  Length:800
##  Class :character
##  Mode  :character
##
##
##
```

```
str(pokemon)
```

```
## 'data.frame':    800 obs. of  9 variables:
##  $ Total     : int  318 405 525 625 309 405 534 634 634 314 ...
##  $ HP        : int  45 60 80 80 39 58 78 78 78 44 ...
##  $ Attack    : int  49 62 82 100 52 64 84 130 104 48 ...
##  $ Defense   : int  49 63 83 123 43 58 78 111 78 65 ...
##  $ Sp..Atk   : int  65 80 100 122 60 80 109 130 159 50 ...
##  $ Sp..Def   : int  65 80 100 120 50 65 85 85 115 64 ...
##  $ Speed     : int  45 60 80 80 65 80 100 100 100 43 ...
##  $ Generation: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Legendary : chr  "False" "False" "False" "False" ...
```

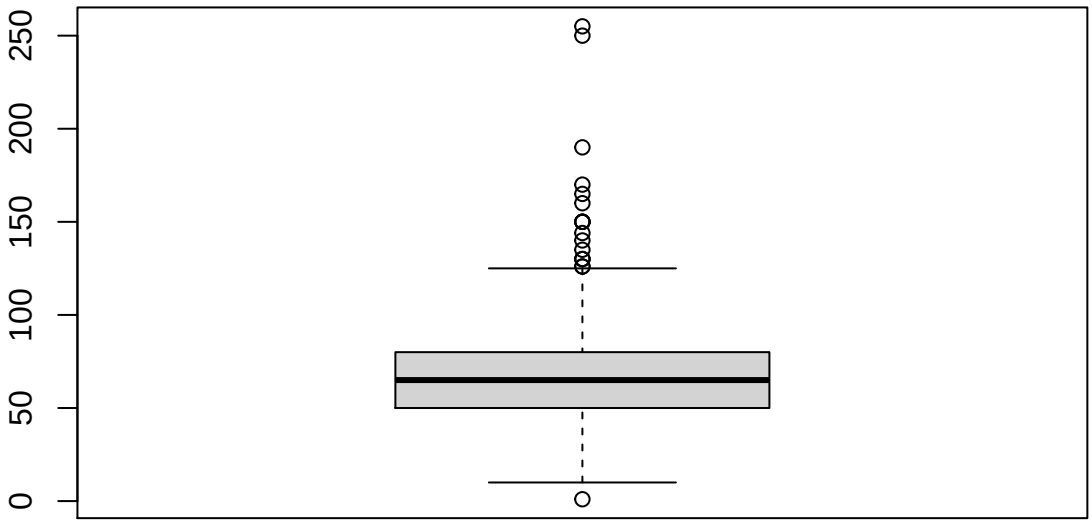
```
#dimensions of the data
dim(pokemon)
```

```
## [1] 800   9
```

```
#check the missing values of data
if(length(which(is.na(pokemon)==T))){
  print("missing values are found")
}else{
  print("no missing values are found")
}
```

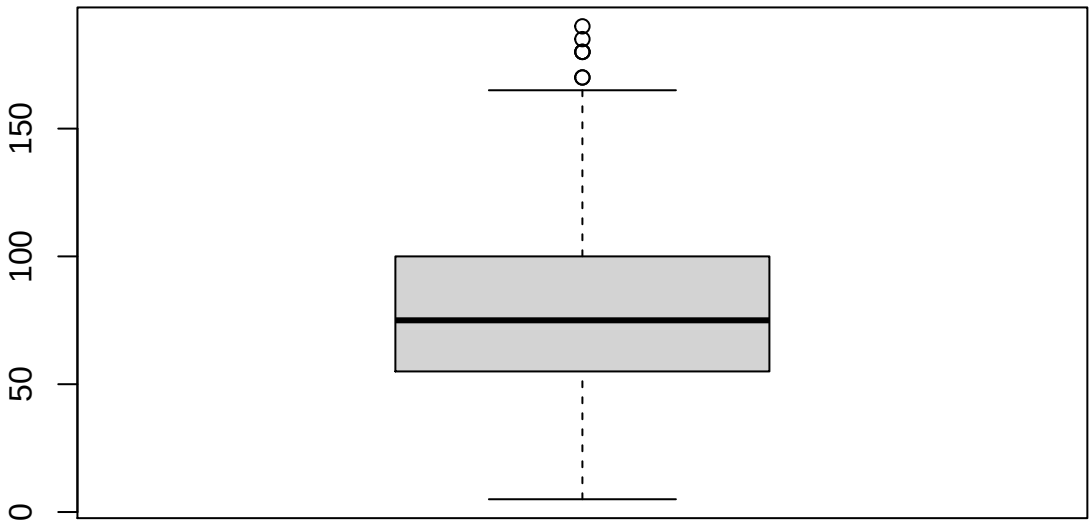
```
## [1] "no missing values are found"
```

```
#check the outliers of the data

unique(boxplot(pokemon$HP)$out)
```



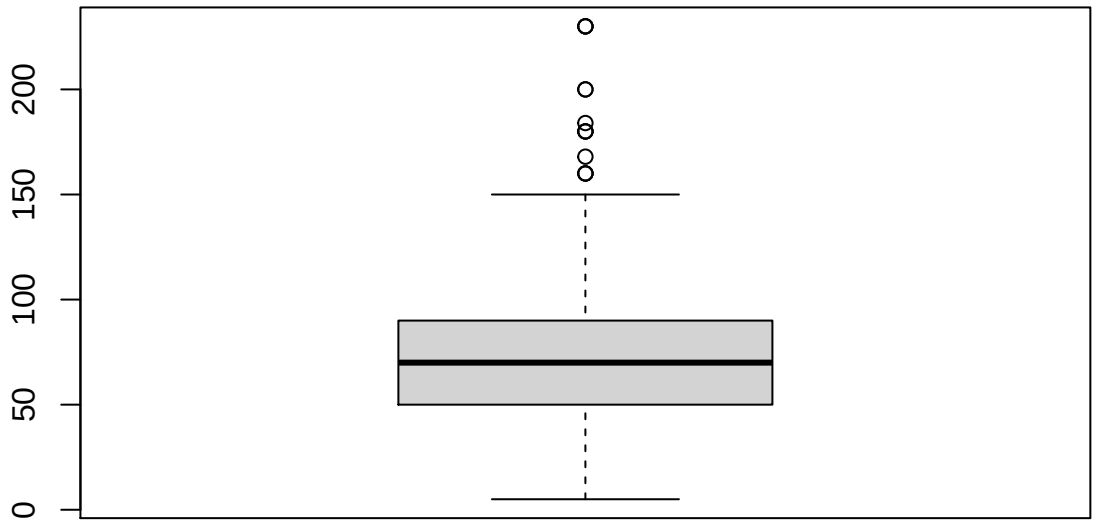

```
##  [1] 140 250 130 160 190 255 150   1 144 170 135 165 126
```

```
unique(boxplot(pokemon$Attack)$out)
```



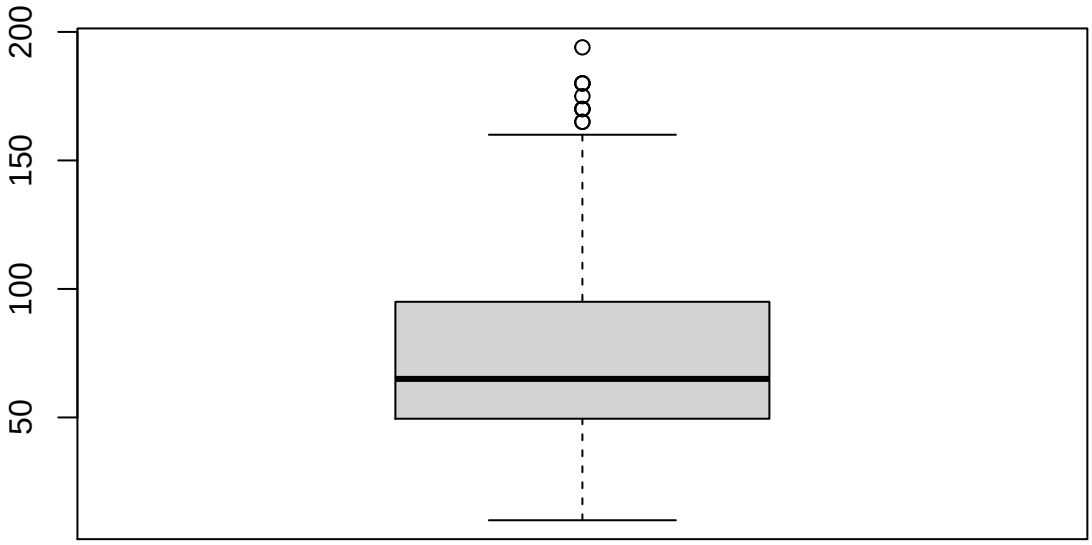

```
## [1] 190 185 180 170
```

```
boxplot(pokemon$Defense)$out
```



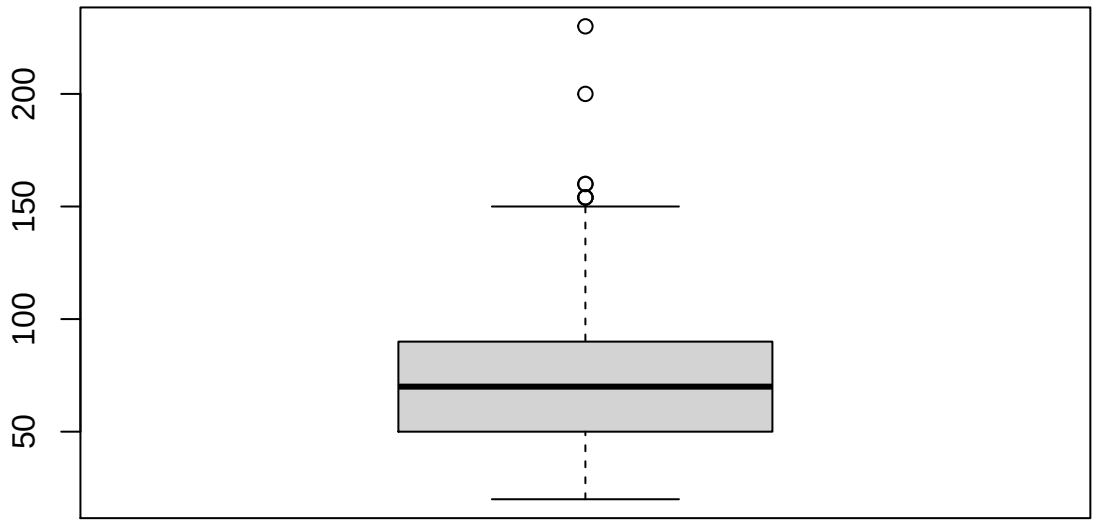

```
##  [1] 180 180 160 200 230 230 180 230 200 160 160 168 184
```

```
boxplot(pokemon$Sp..Atk)$out
```



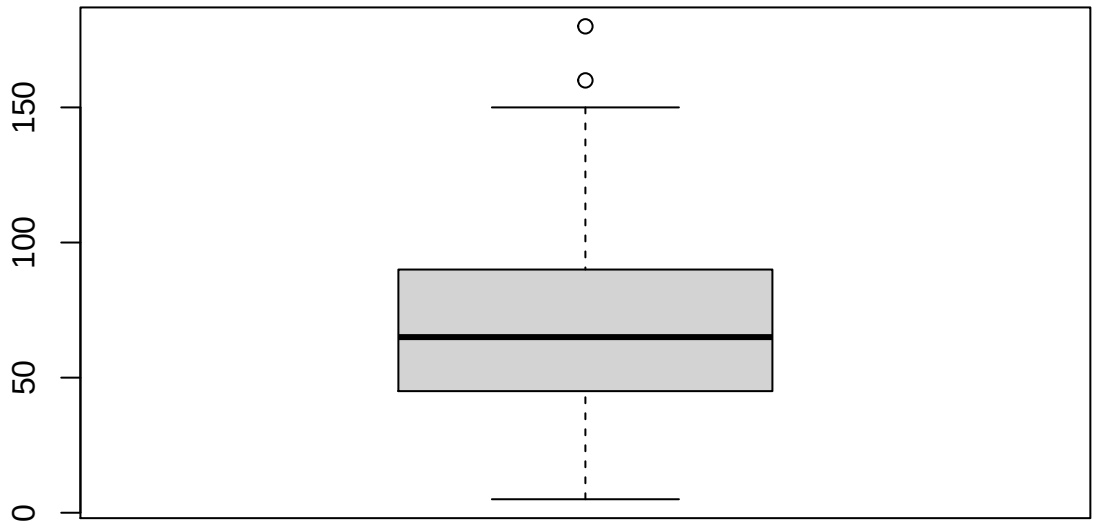

```
##  [1] 175 170 194 165 165 180 180 180 170 170
```

```
boxplot(pokemon$Sp..Def)$out
```



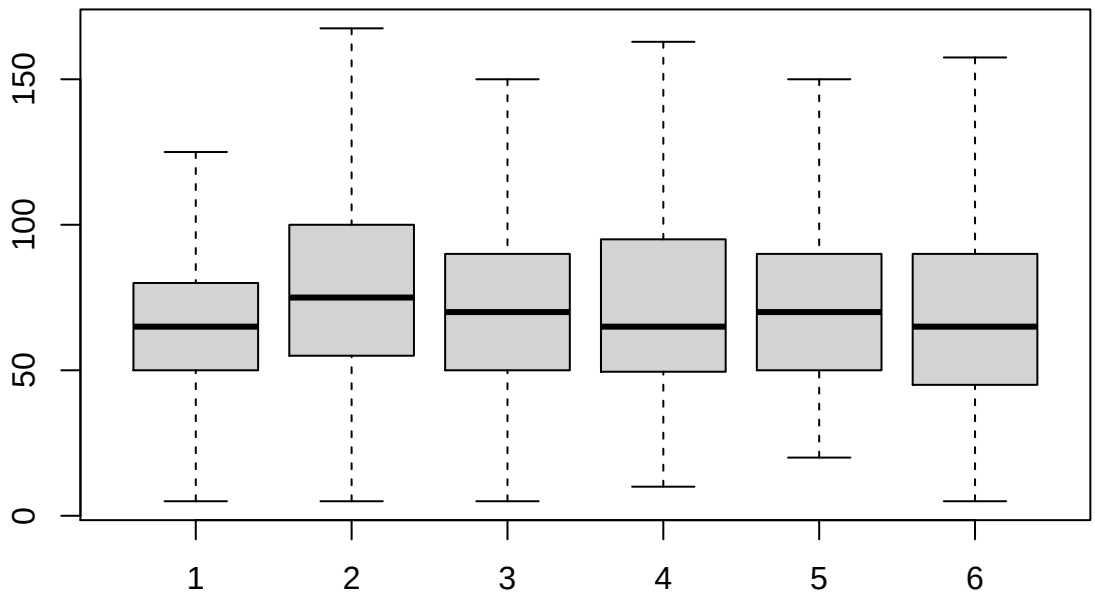

```
## [1] 230 154 154 200 160 160 154
```

```
boxplot(pokemon$Speed)$out
```




```
## [1] 160 180
```

```
#outliers replace with mean
for (colName in c('HP','Attack','Defense','Sp..Atk','Sp..Def','Speed')) {
  high = quantile(pokemon[,colName])[4] + 1.5*IQR(pokemon[,colName])
  low = quantile(pokemon[,colName])[2] - 1.5*IQR(pokemon[,colName])
  for (index in c(1:nrow(pokemon))) {
    pokemon[,colName][index] = ifelse(pokemon[,colName][index] > high, high, pokemon[,colName][index])
    pokemon[,colName][index] = ifelse(pokemon[,colName][index] < low, low, pokemon[,colName][index])
  }
}
#after removing Outliers boxplot
boxplot(pokemon$HP,pokemon$Attack,pokemon$Defense,pokemon$Sp..Atk,pokemon$Sp..Def,pokemon$Speed)

# Load libraries
library(ggplot2)
```




```
library(tidyverse)
```

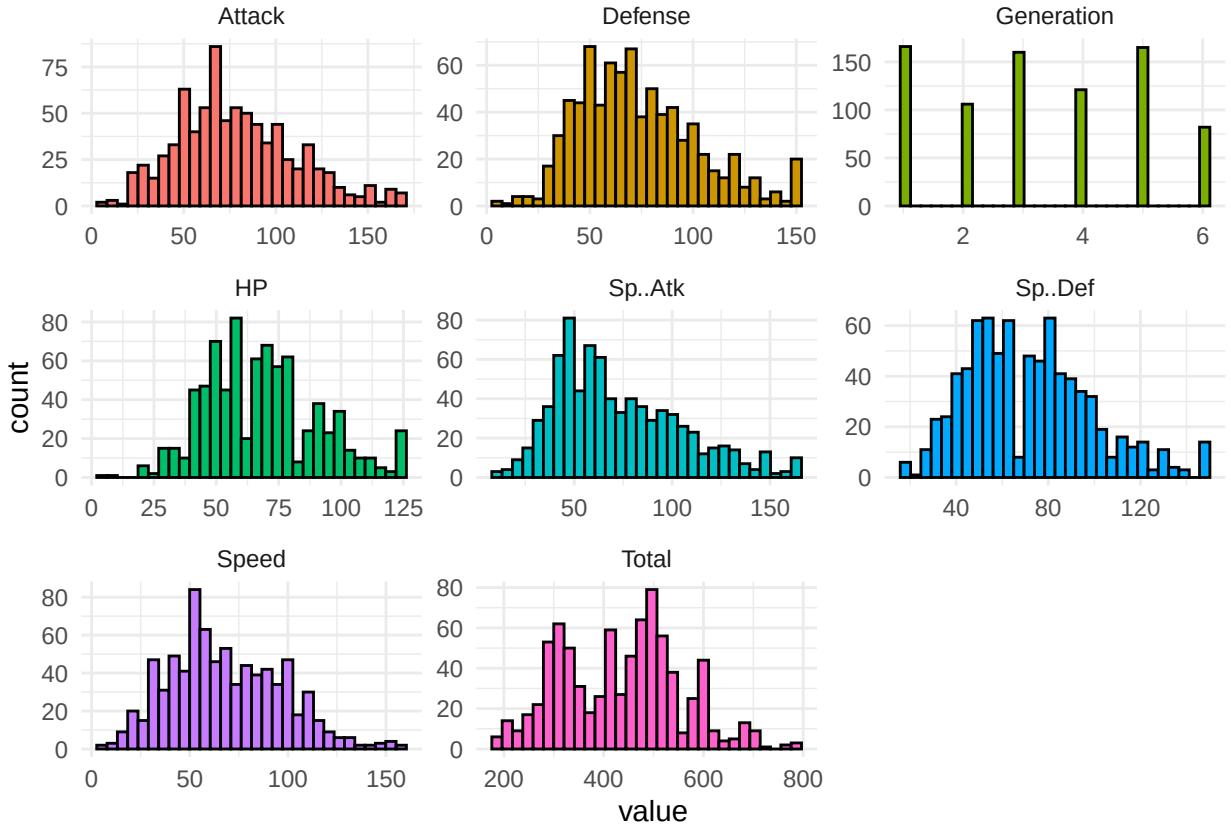
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra)
```

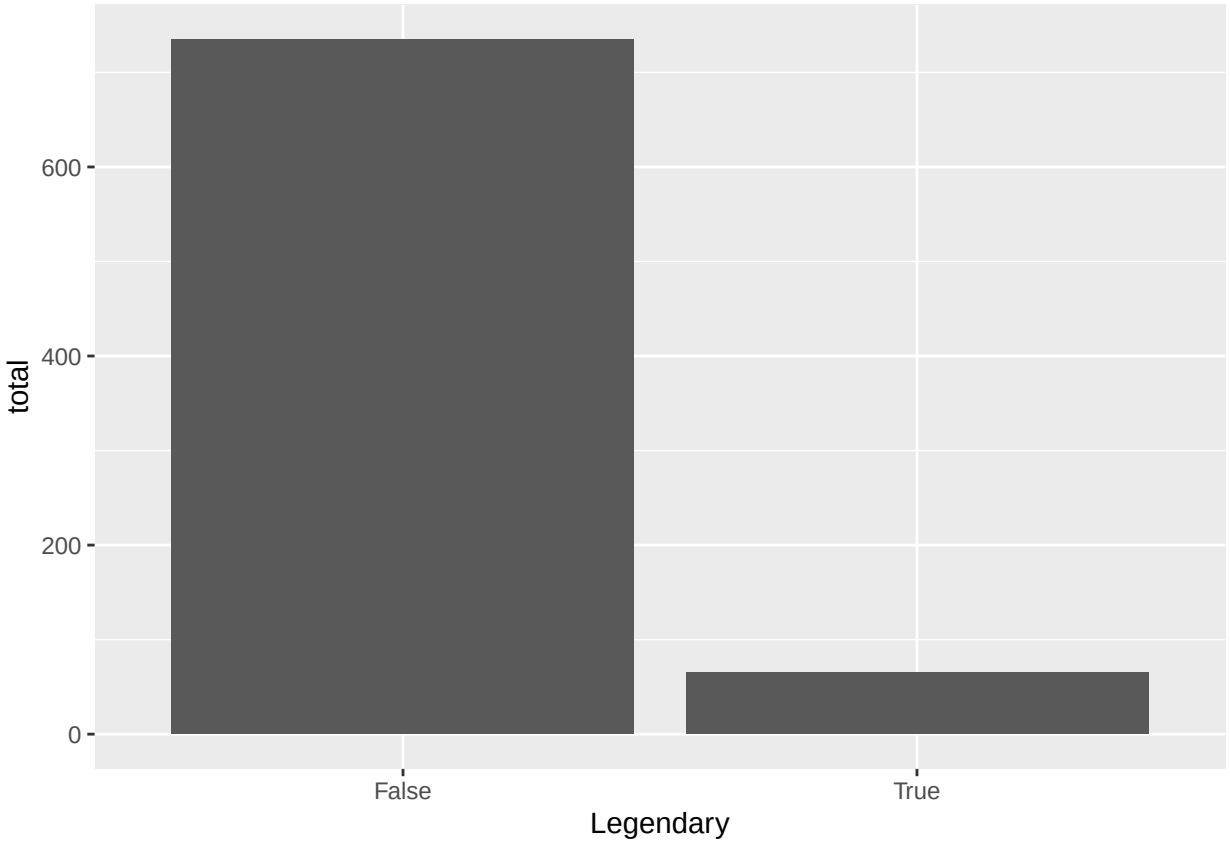
```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(knitr)

#histogram plots of pokemon
library(tidyverse)
pokemon%>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot() +
  geom_histogram(mapping = aes(x=value,fill=key), color="black") +
  facet_wrap(~ key, scales = "free") +
  theme_minimal() +
  theme(legend.position = 'none')
```

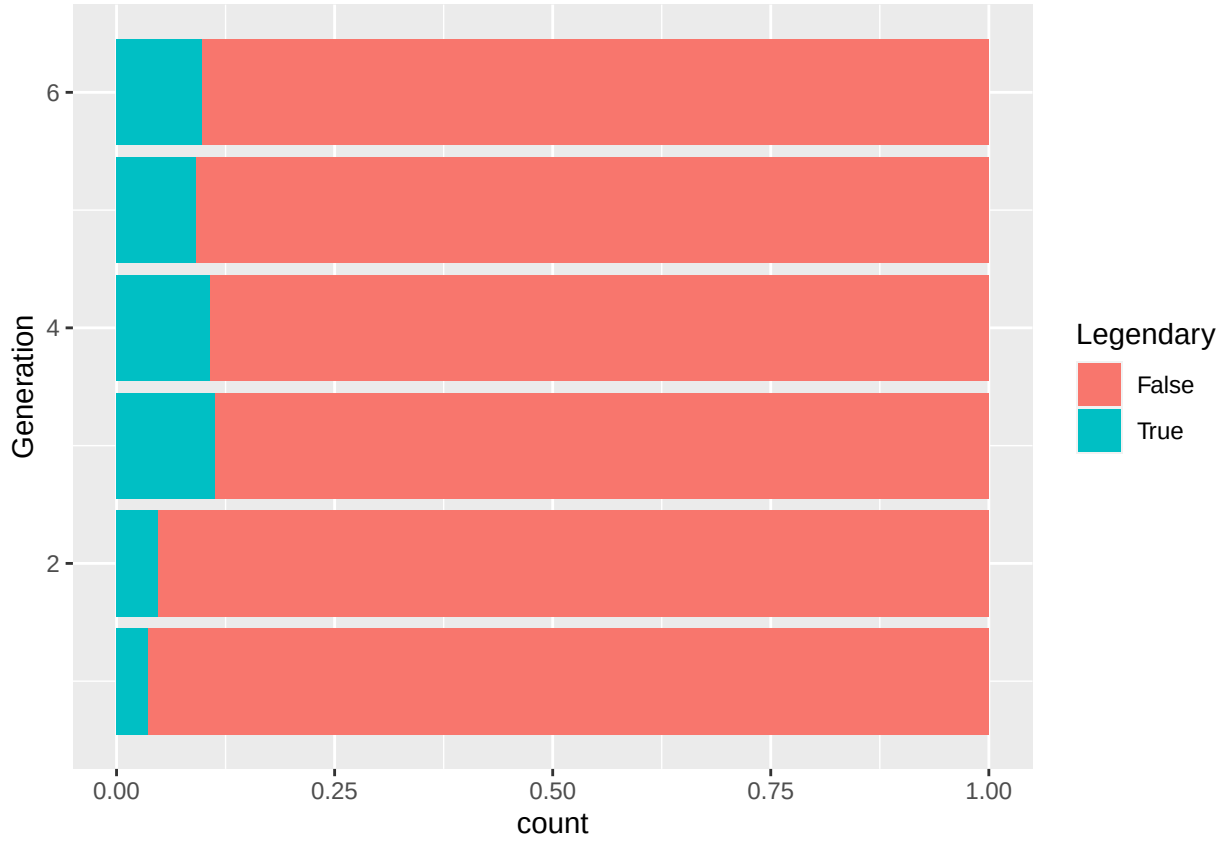
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#ggplot for total and legendary variables
ggplot(pokemon,aes(x =Legendary, fill =Total)) +
  geom_bar(position = "stack")+labs(y="total")
```




```
#ggplot for input(generation) and output(legendary) variable
ggplot(pokemon, aes(Generation)) +
  geom_bar(aes(fill =Legendary), position = "fill") +
  coord_flip()
```

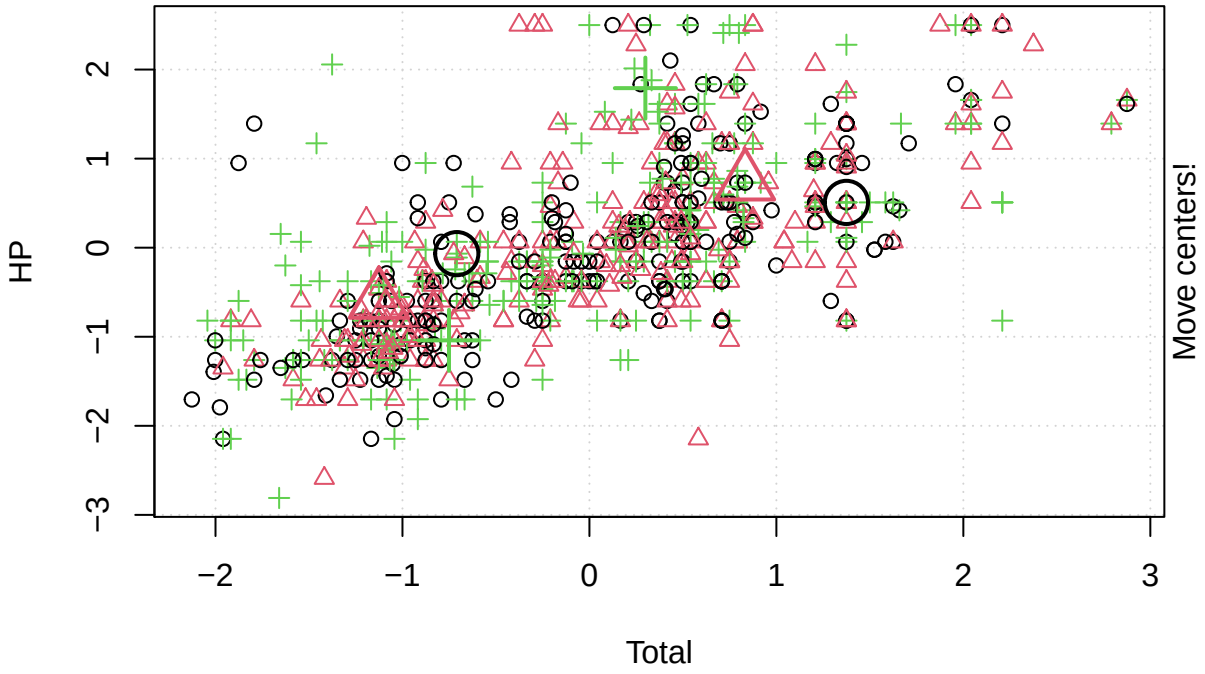


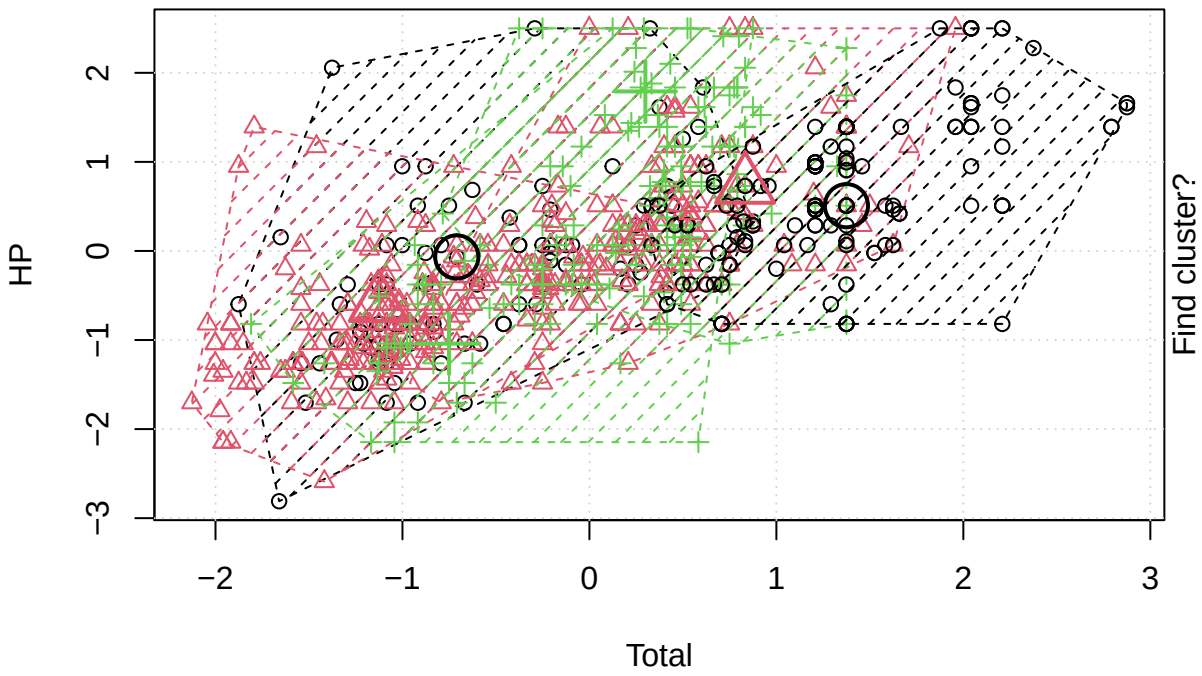

```
      #NOTE:I am observing 3rd generation pokemons more legendarys compare to other generations.

#categorical in to numerical
pokemon$Legendary=as.numeric(factor(pokemon$Legendary))-1

#scale the data
pokemon1=scale(pokemon[1:7])
pokem=pokemon[-1:-7]
pokemonn=cbind(pokemon1,pokem)
# View after normalize the data
View(pokemonn)


############## 1.kmeans clustering ############

km <- kmeans(pokemonn,6) #kmeans clustering
str(km)
```

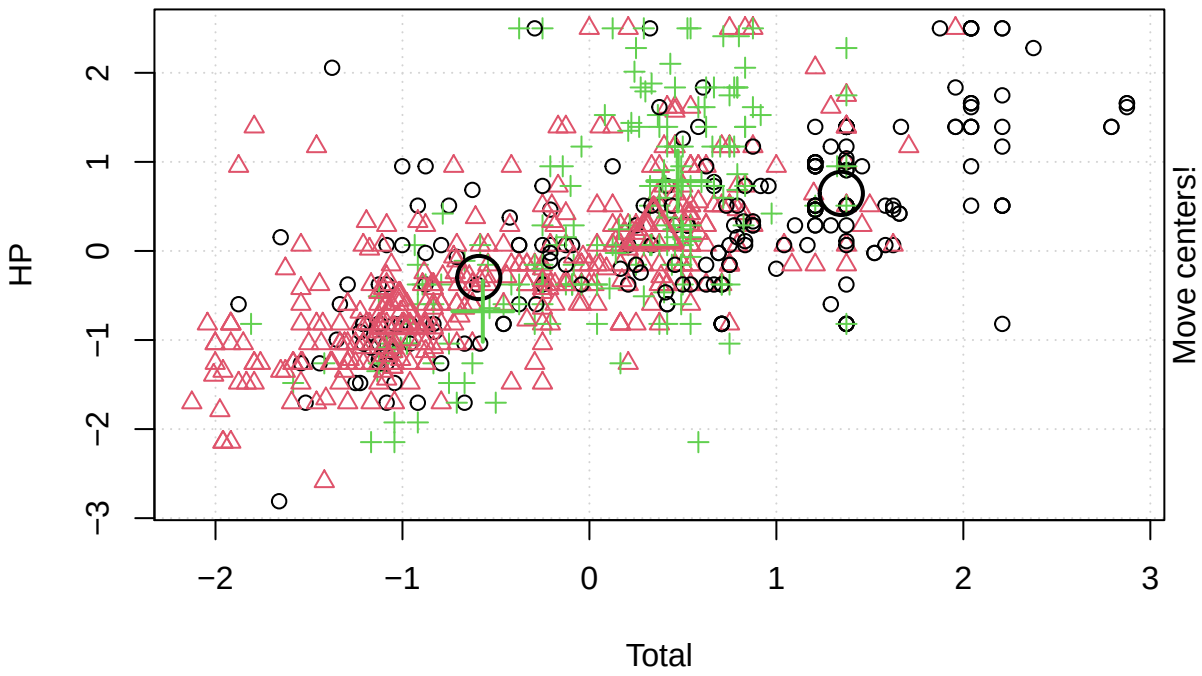
```
## List of 9
##  $ cluster     : int [1:800] 6 1 1 4 6 1 1 4 4 6 ...
##  $ centers     : num [1:6, 1:9] 0.203 -0.928 0.585 1.633 0.598 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:9] "Total" "HP" "Attack" "Defense" ...
##  $ totss       : num 7858
##  $ withinss    : num [1:6] 821 411 399 547 676 ...
```

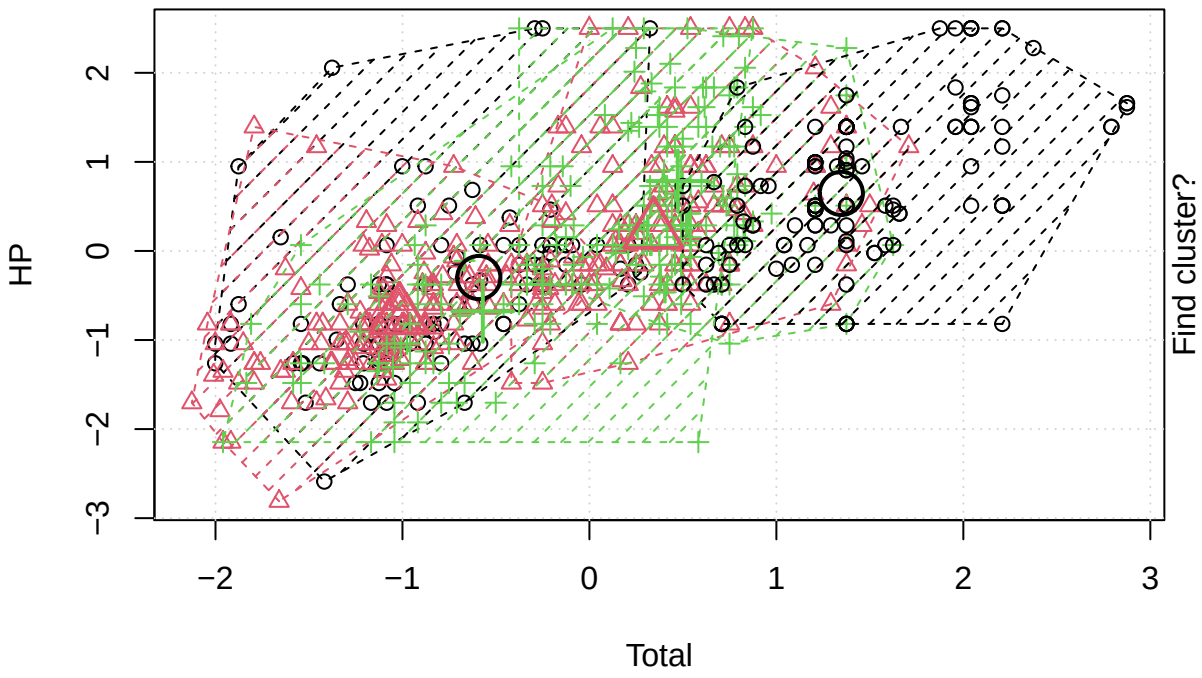
```
##  $ tot.withinss: num 3378
##  $ betweenss   : num 4480
##  $ size        : int [1:6] 186 146 70 88 157 153
##  $ iter        : int 5
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

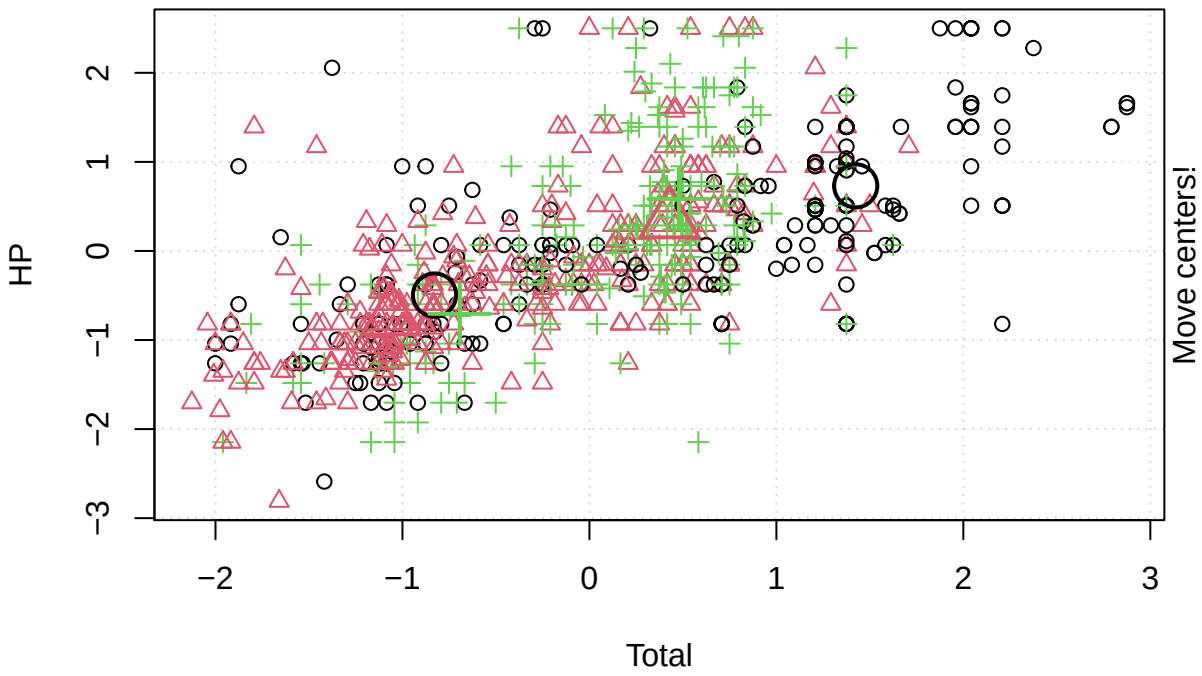
```
# now we have to group by calculating centrioid and finding the distance between
#centroids and variable and group the variable as per least distanc
library(animation)
km=kmeans.ani(pokemonn, 6)
```

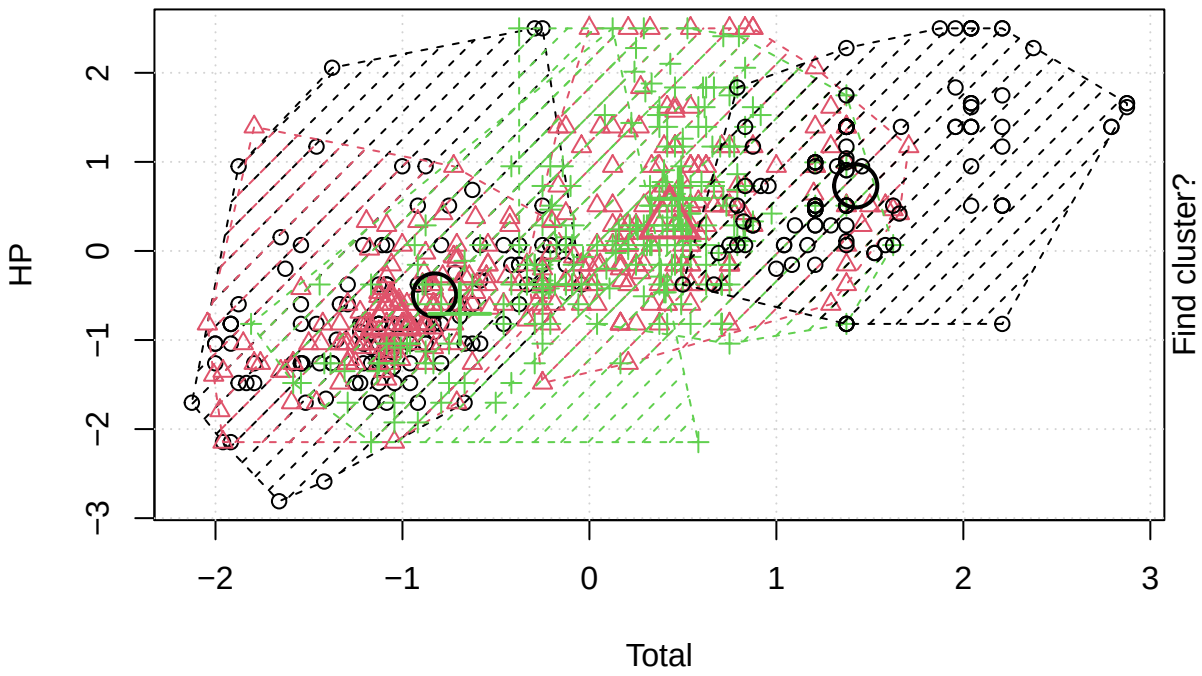


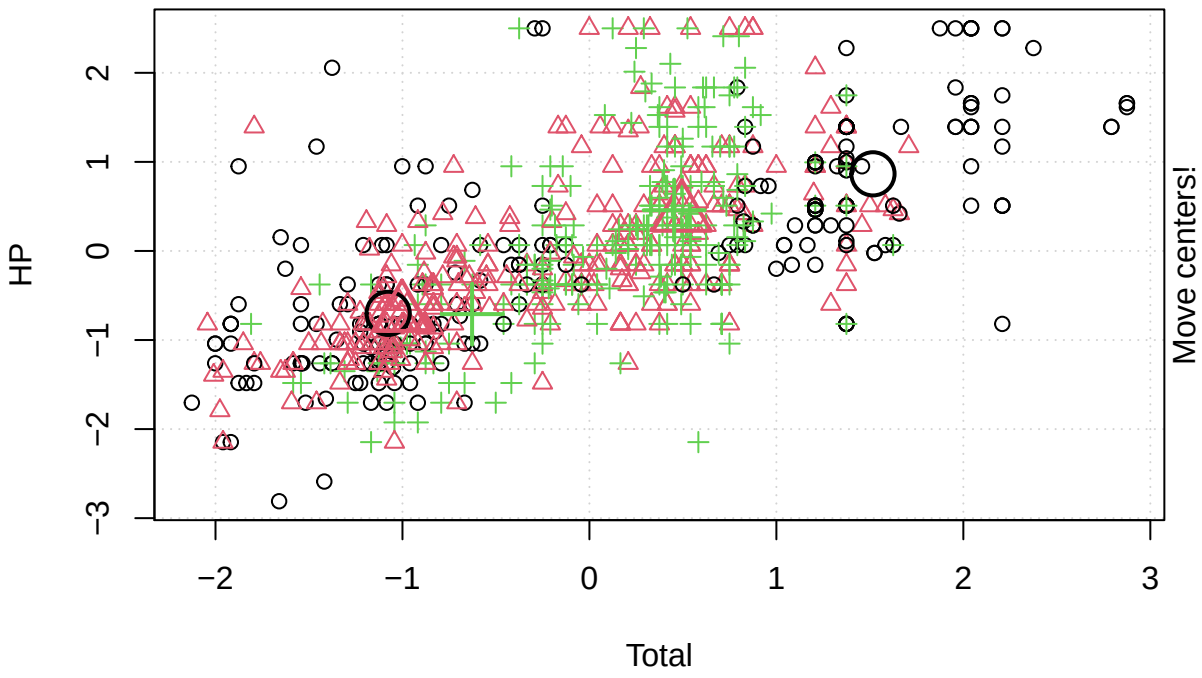

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

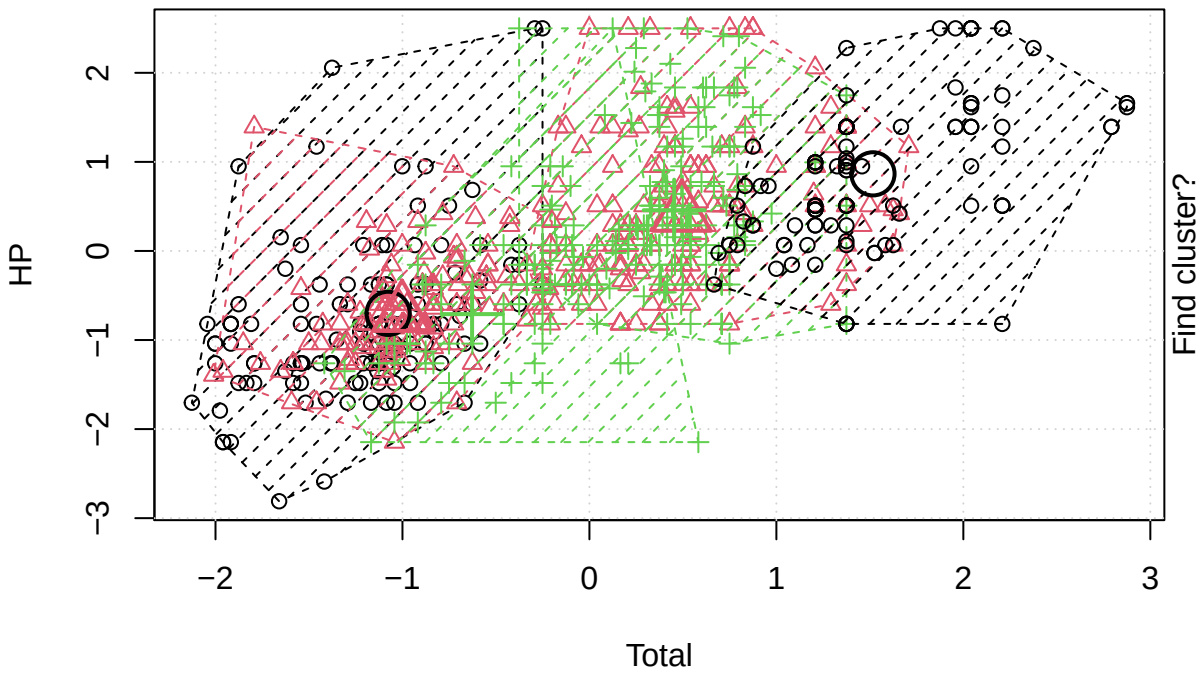

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

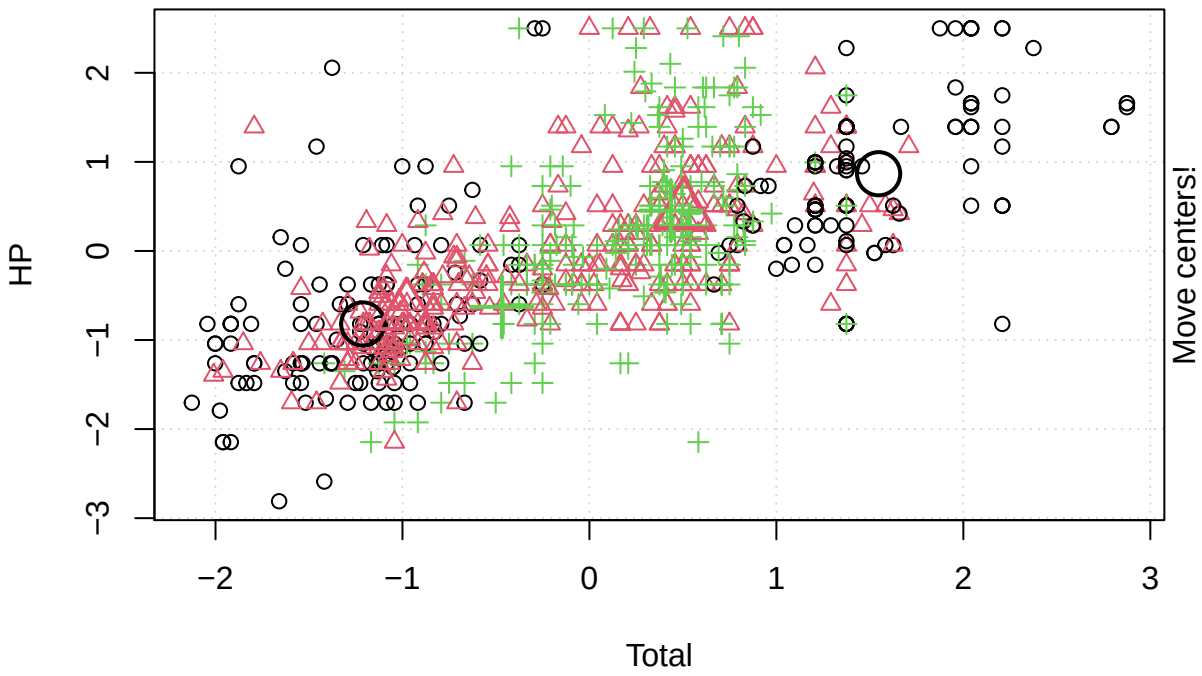

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

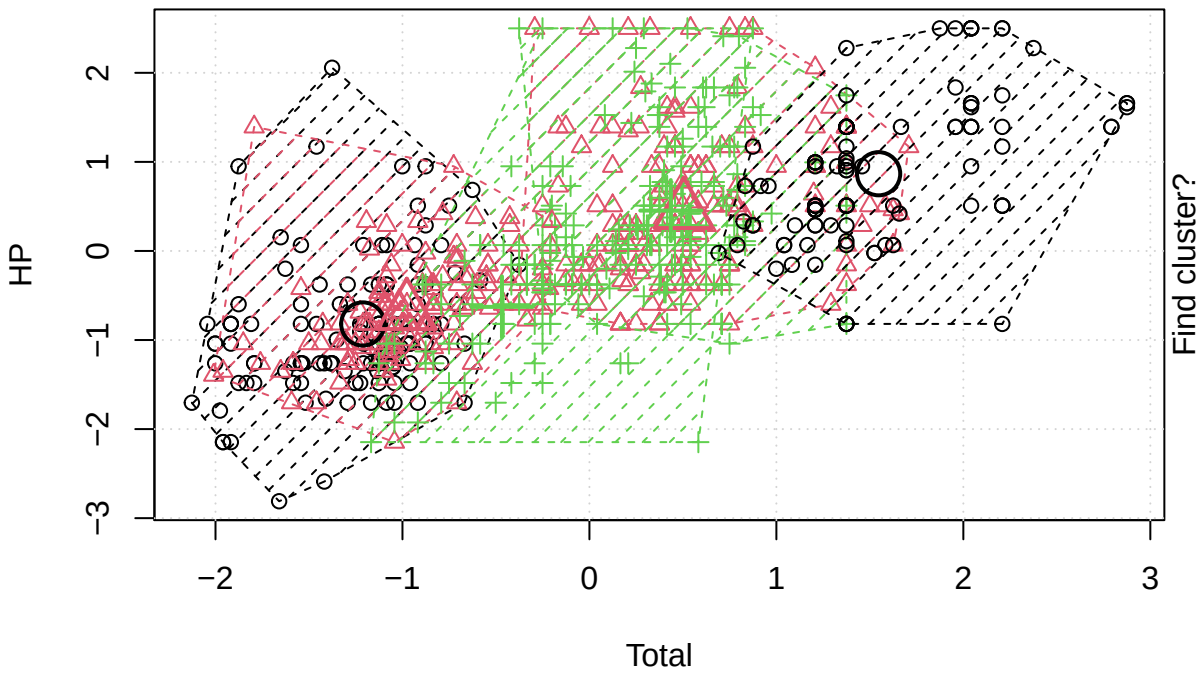

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

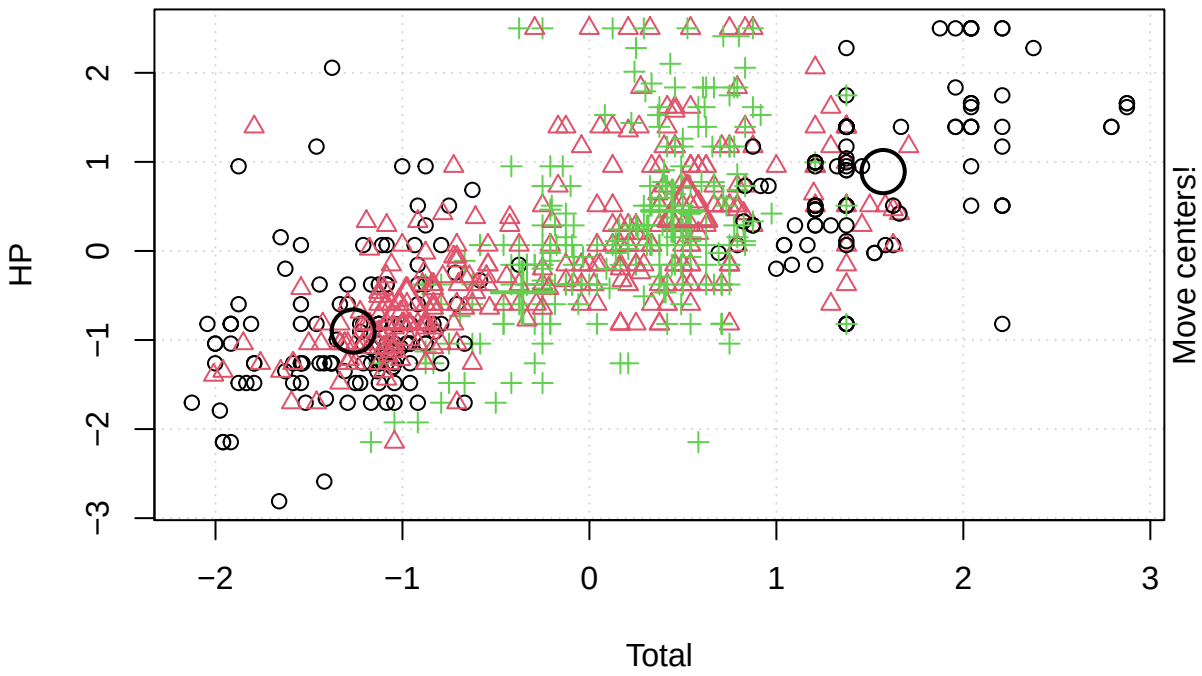

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

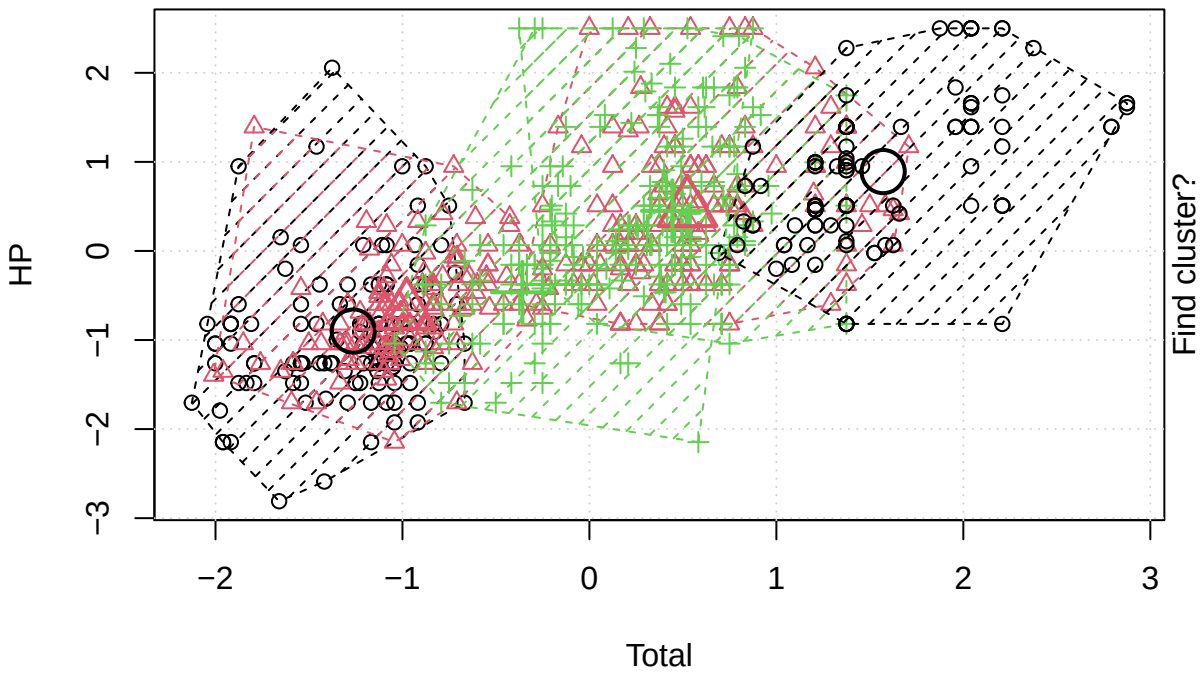

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

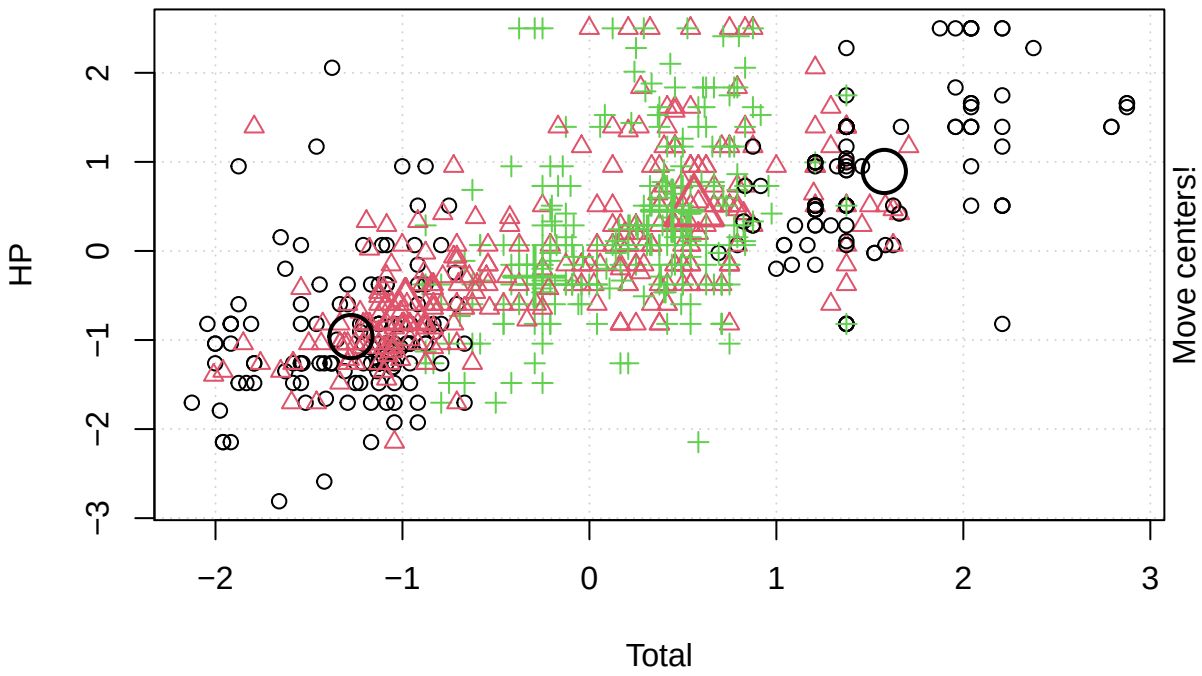

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

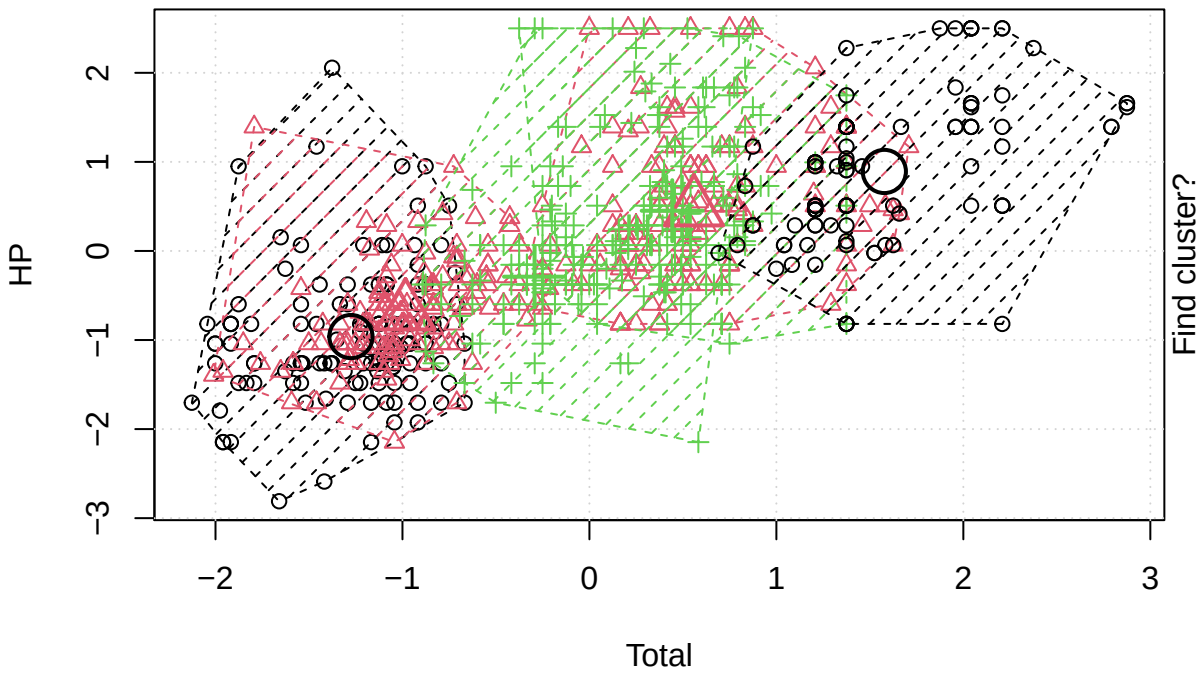

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

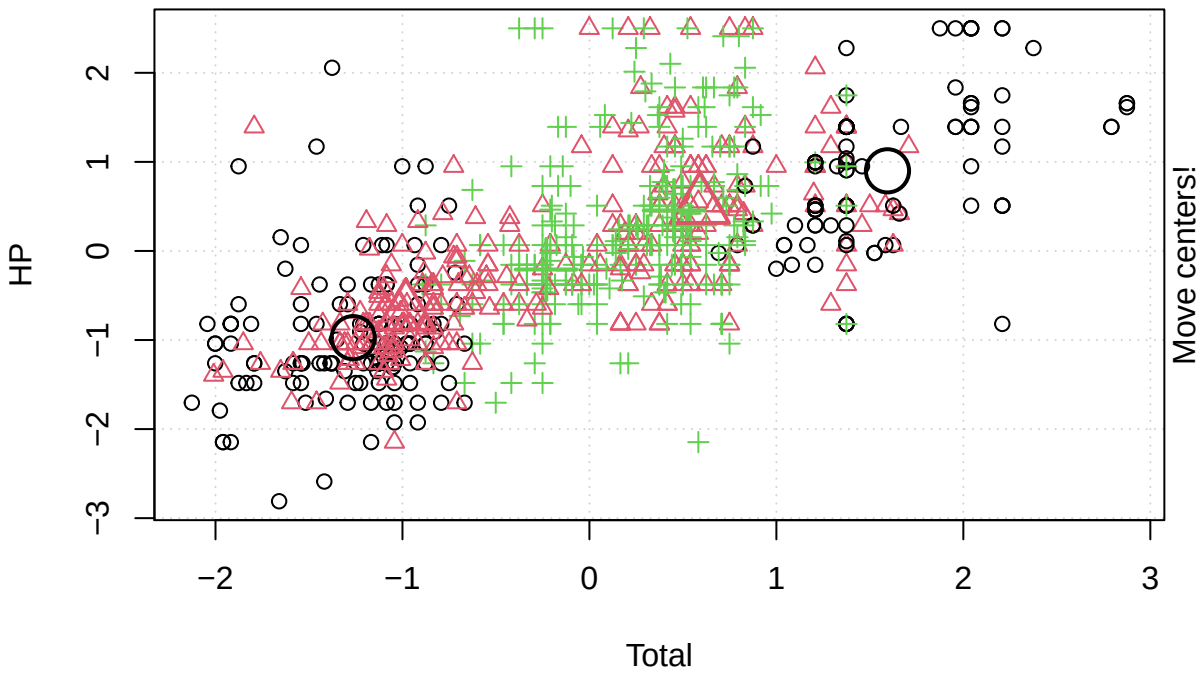

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

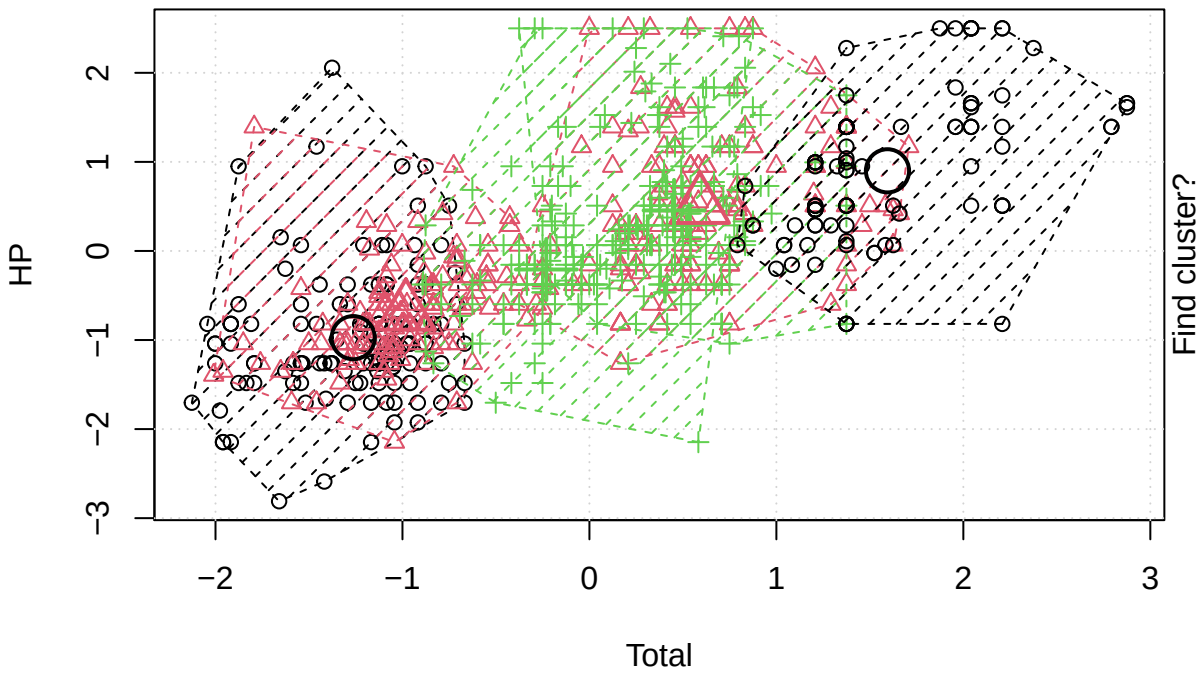

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

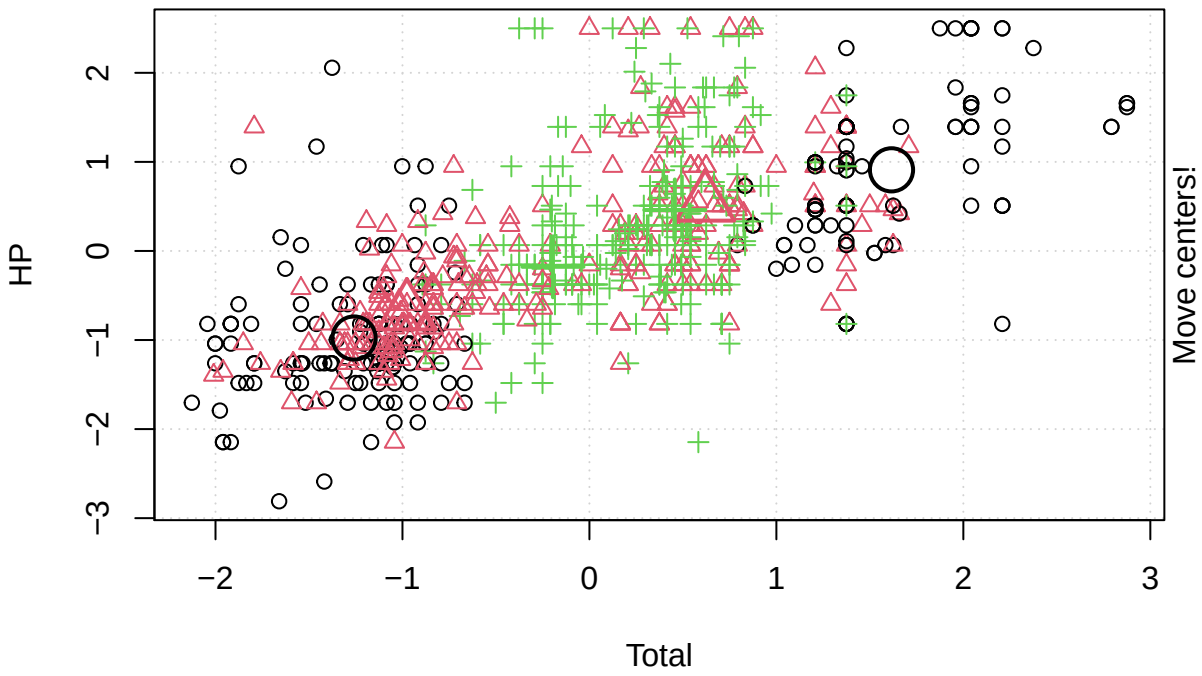

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

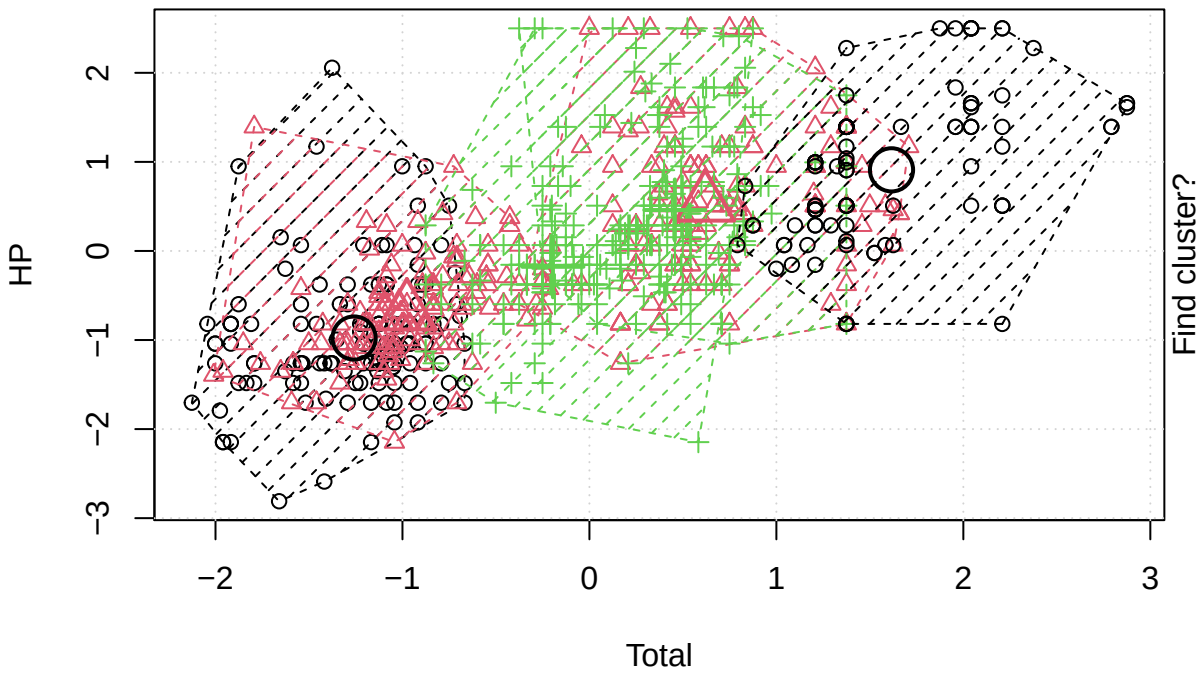

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

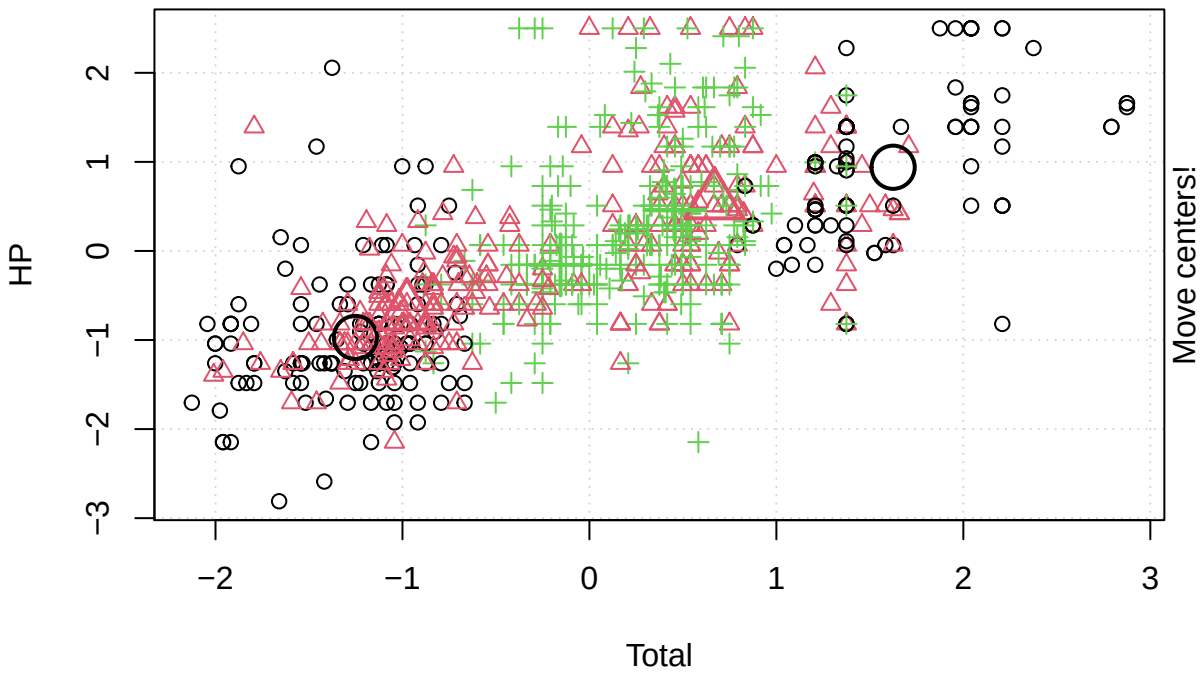

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

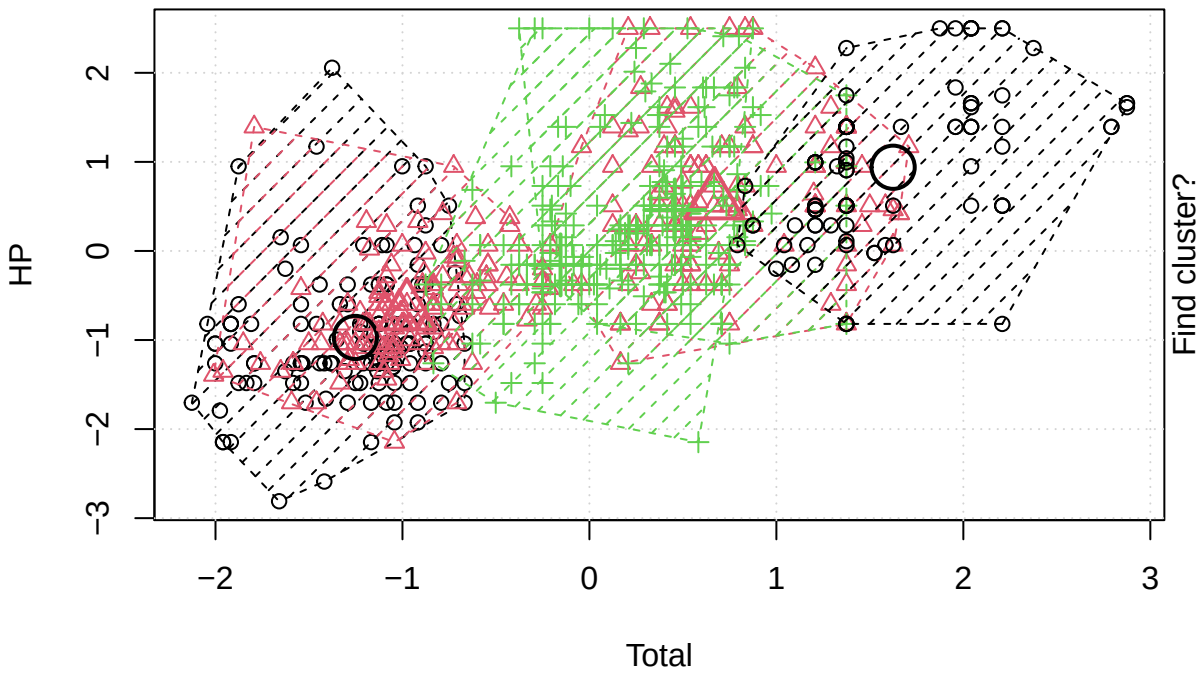

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

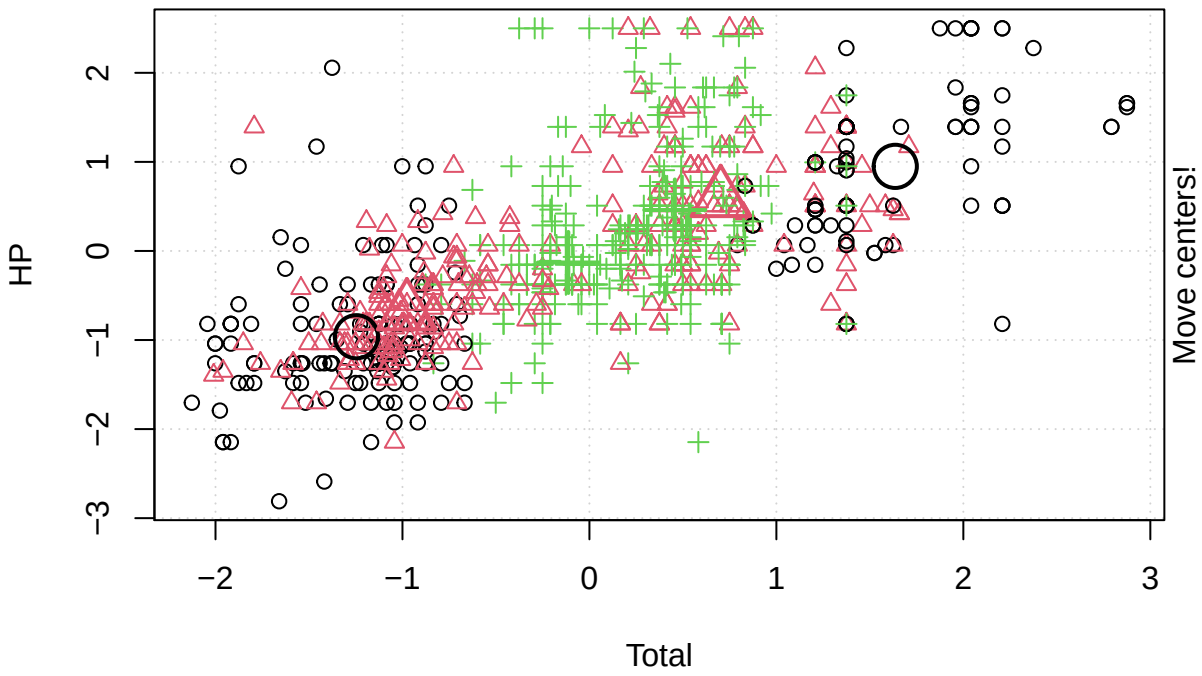

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

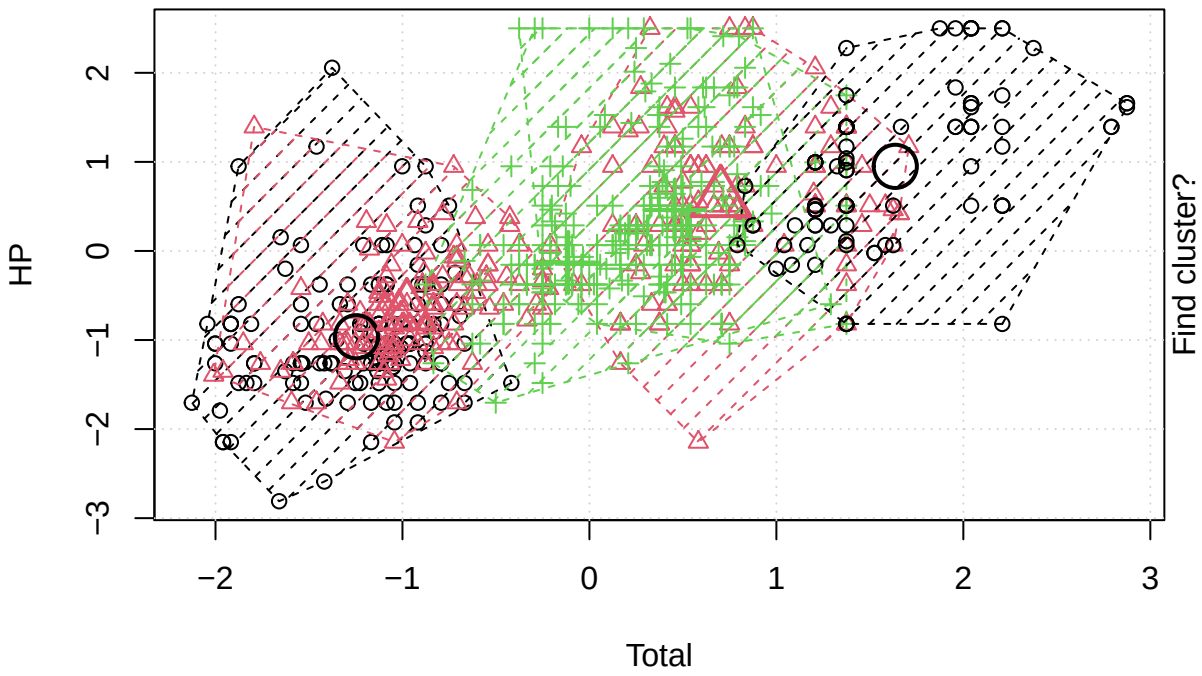

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

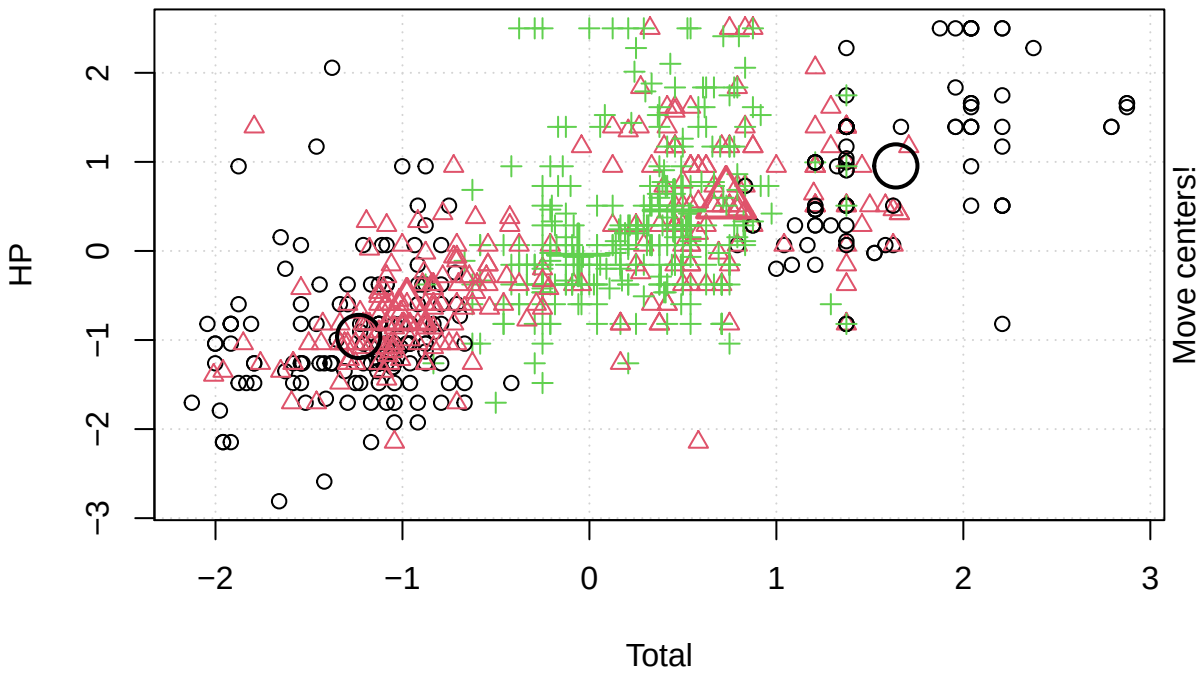

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

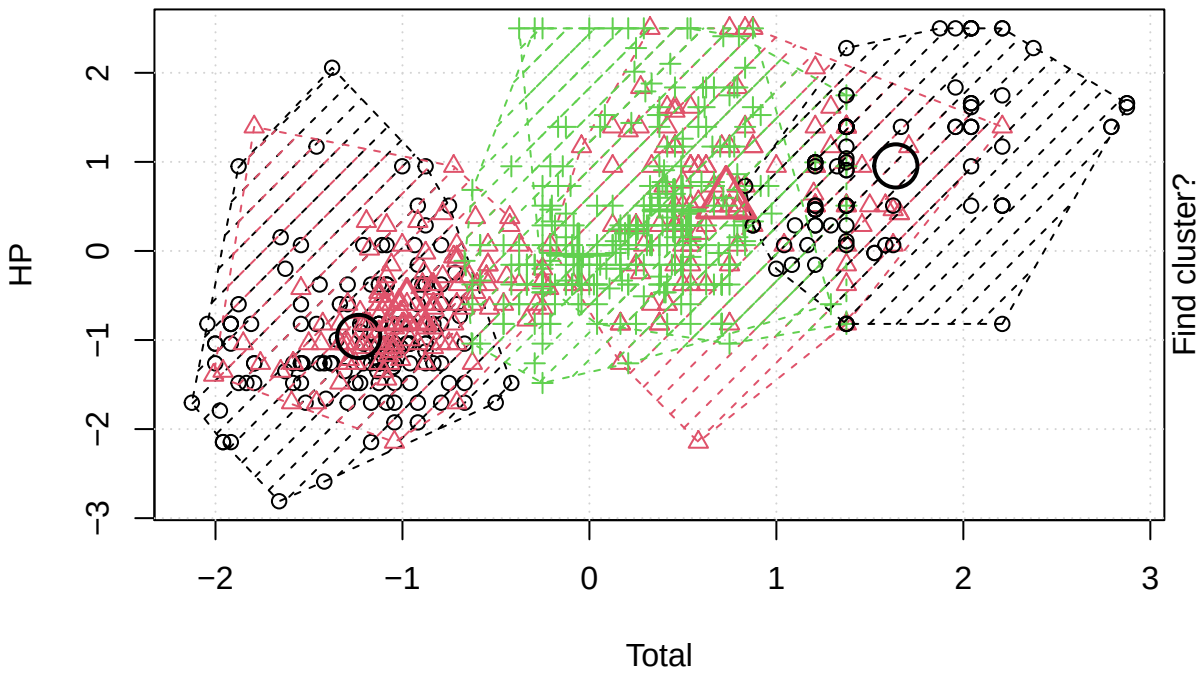

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

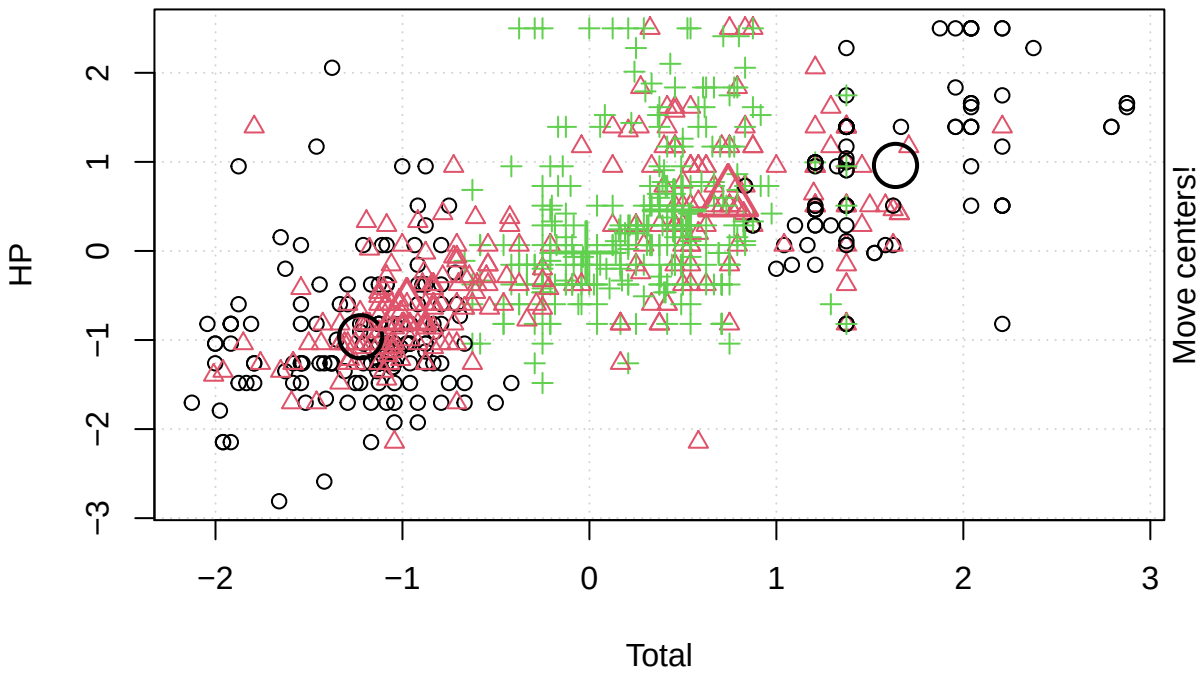

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

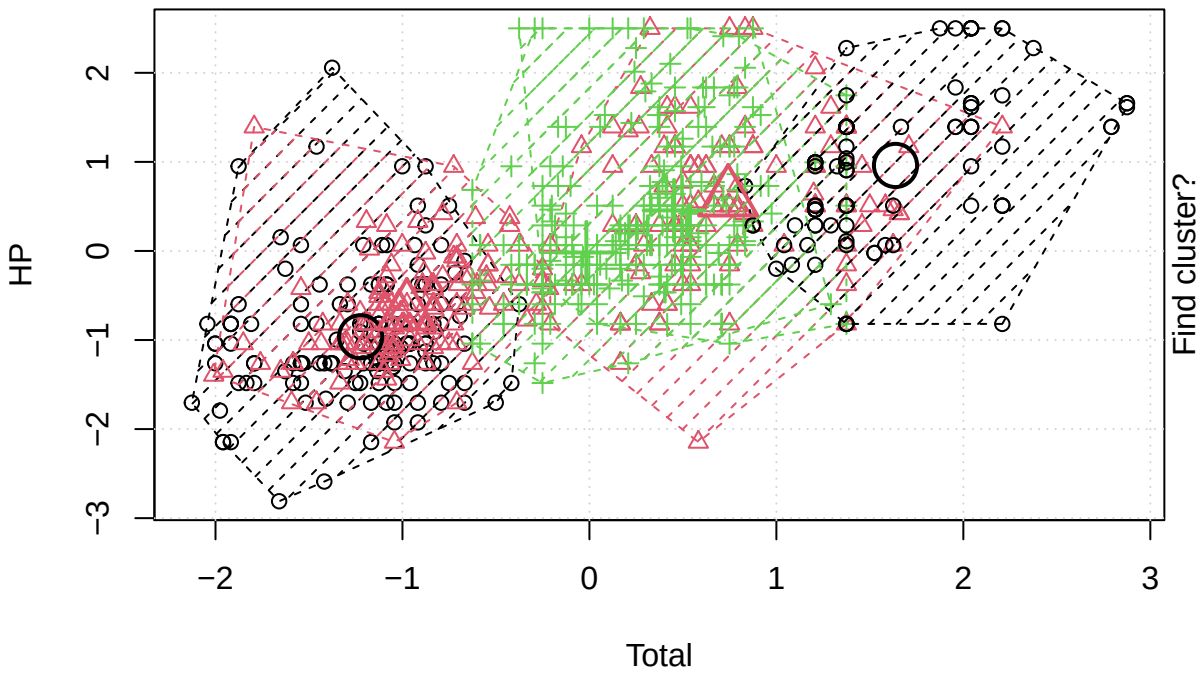

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

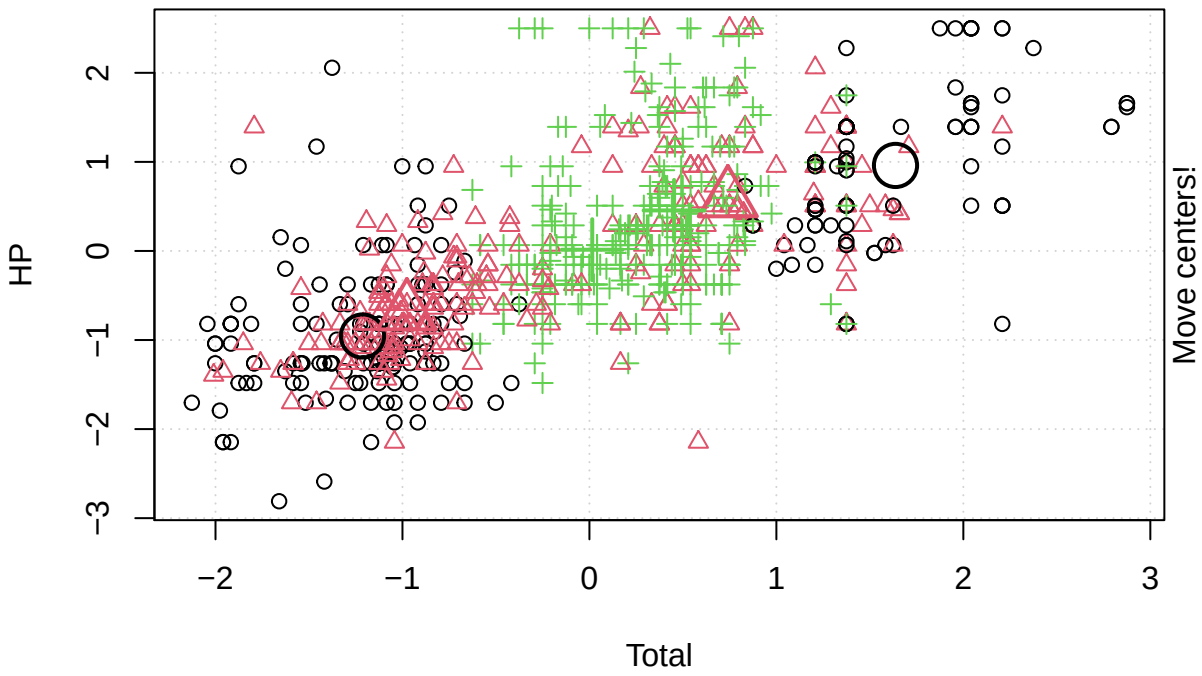

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

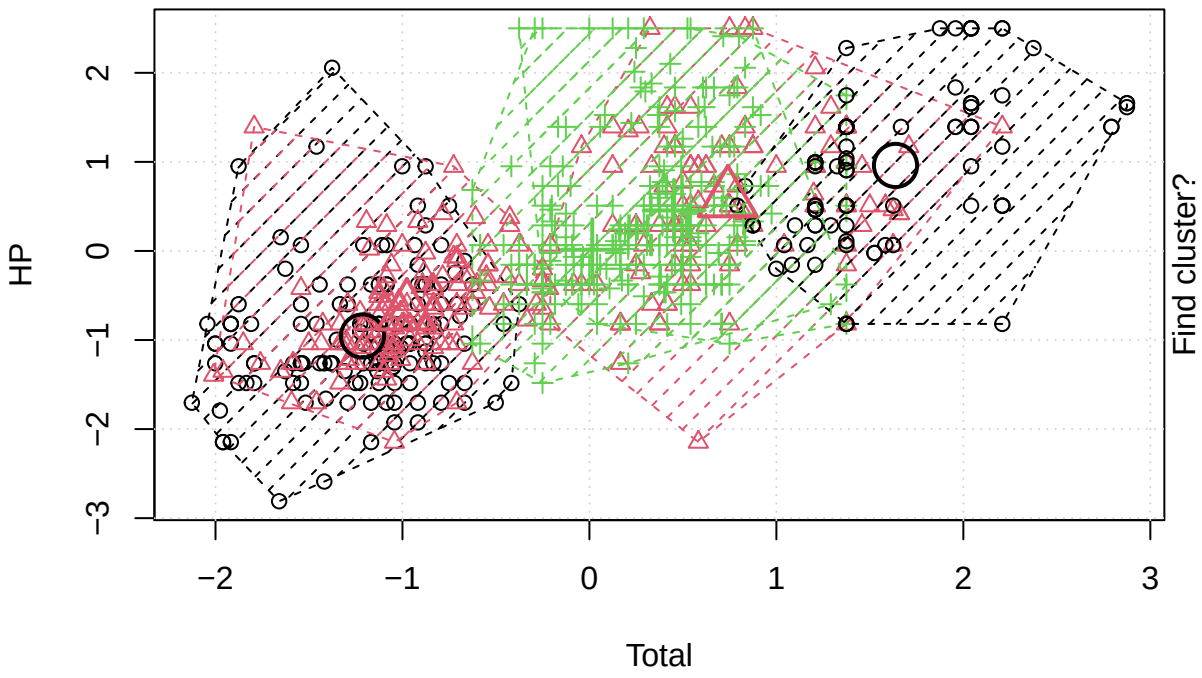

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

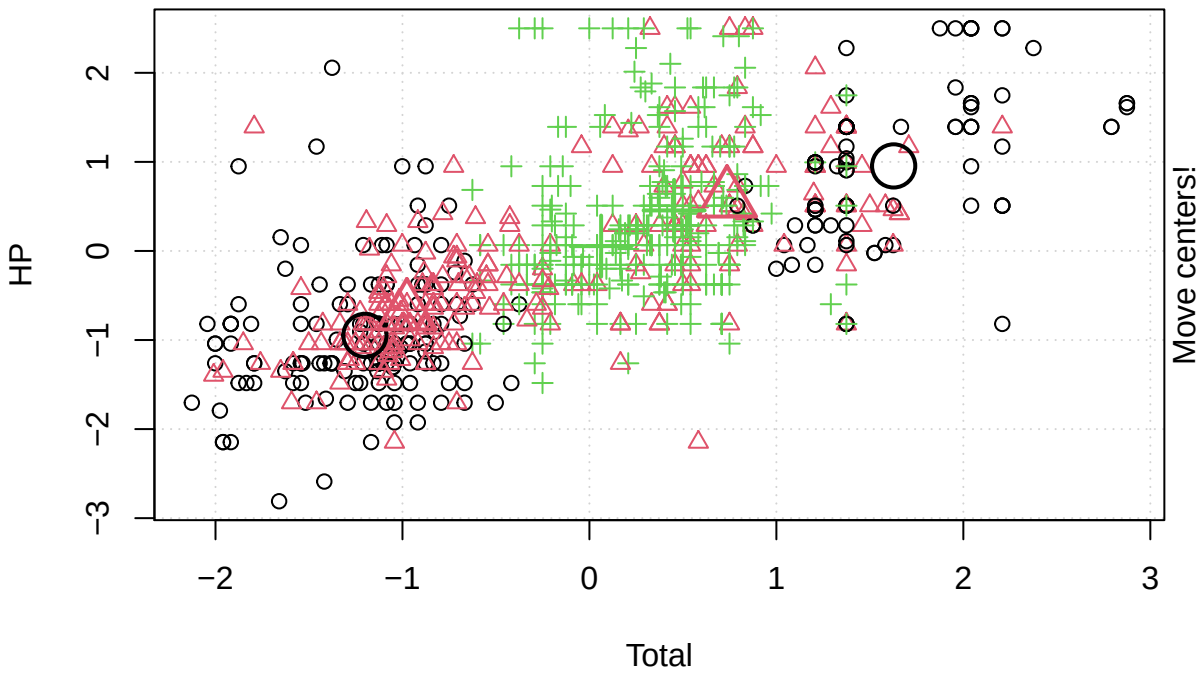

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

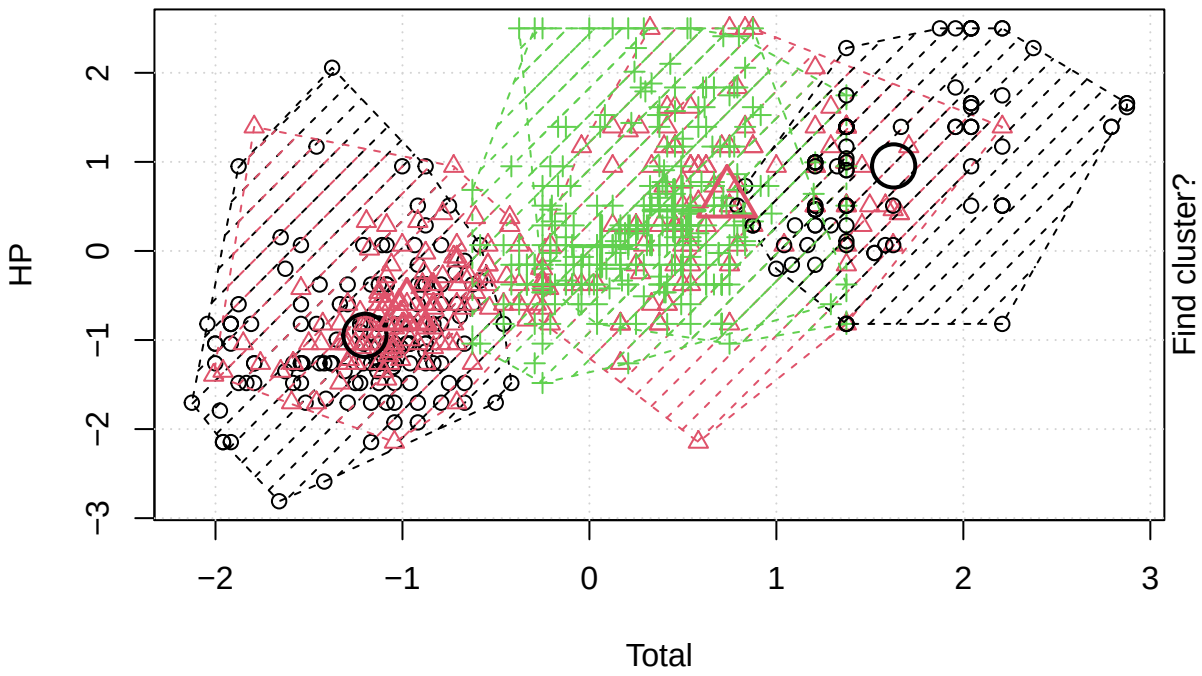

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

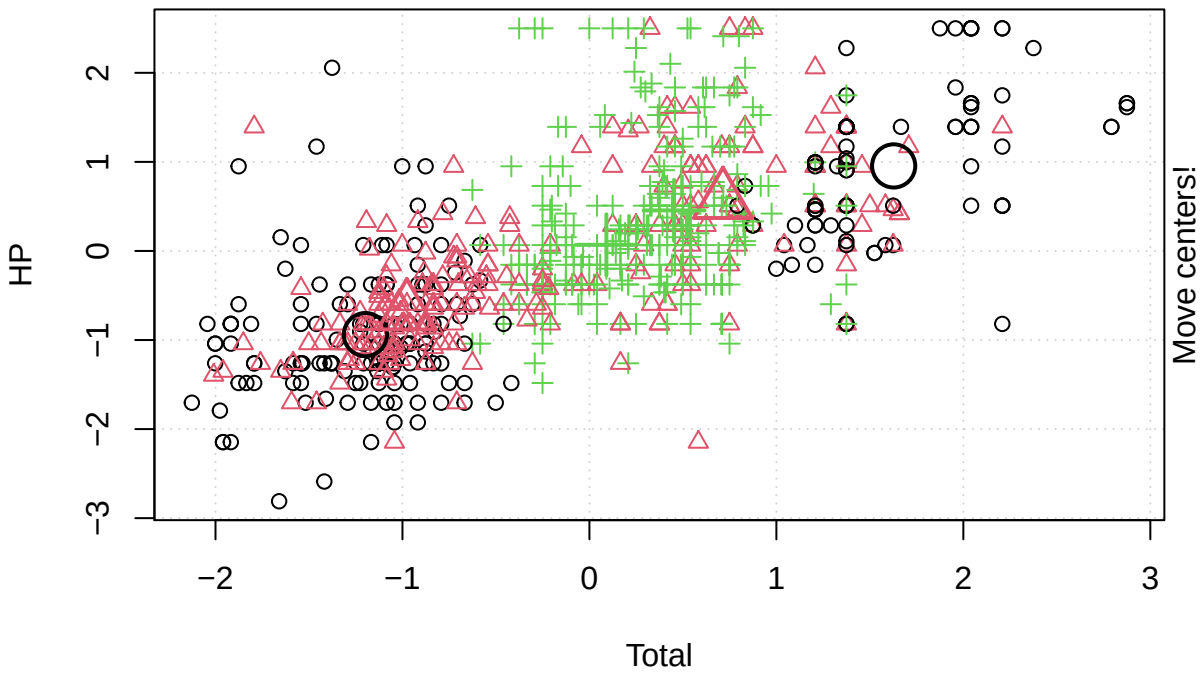

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

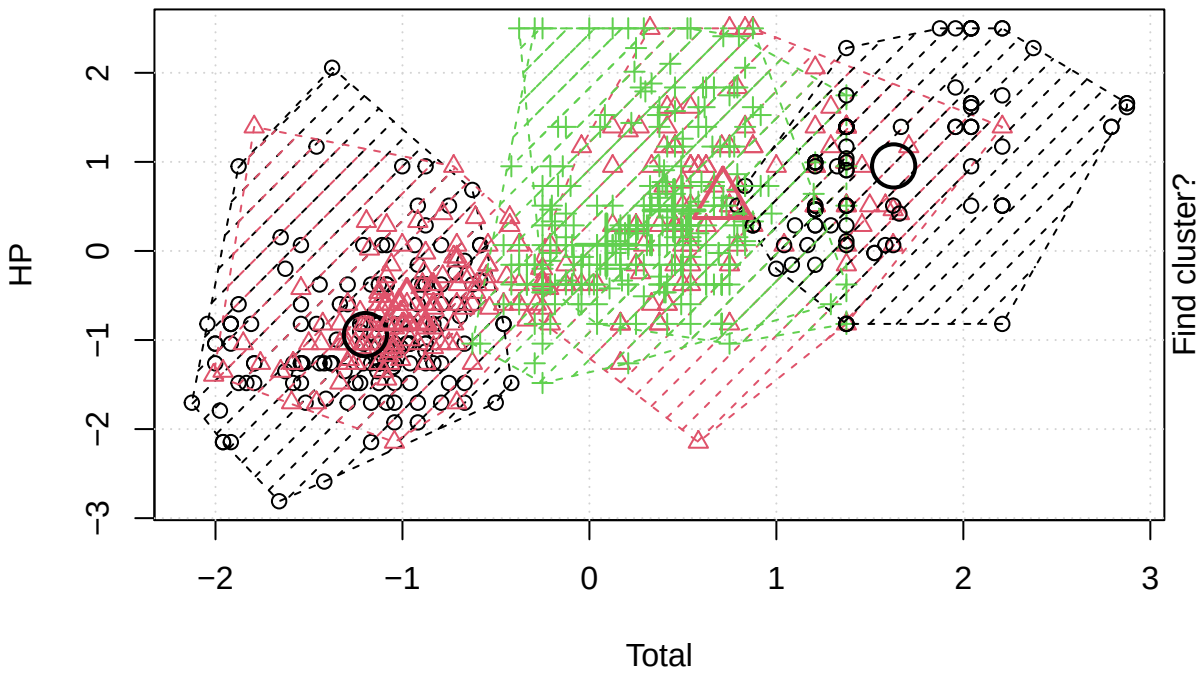

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

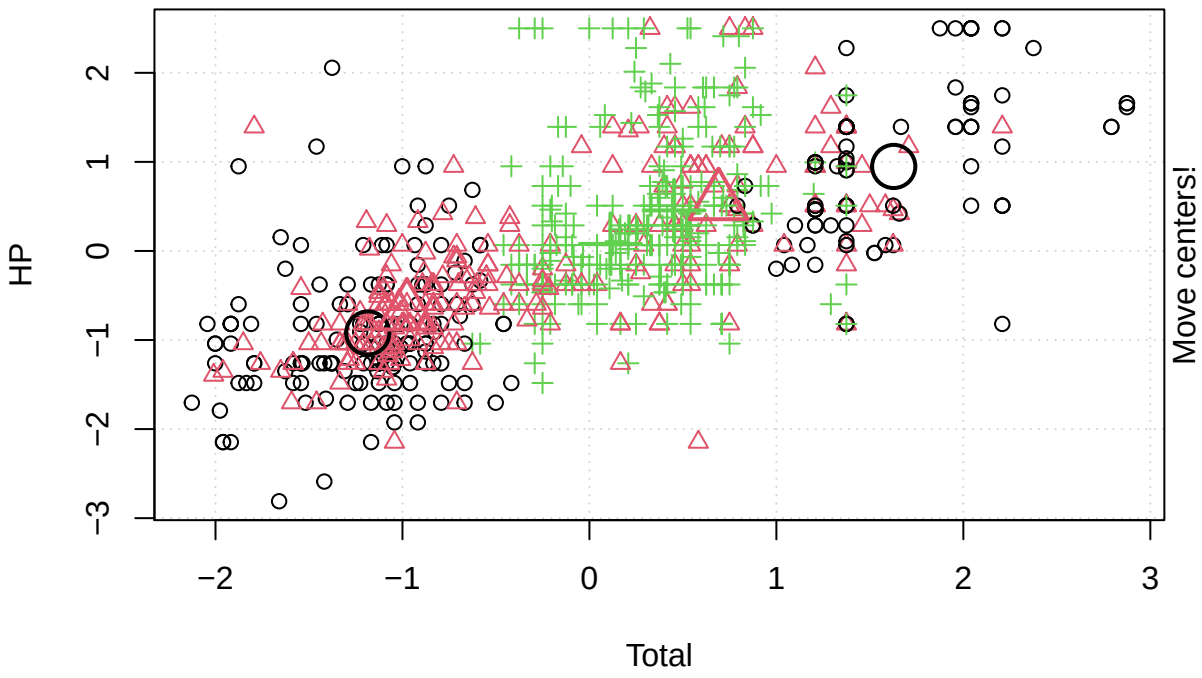

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

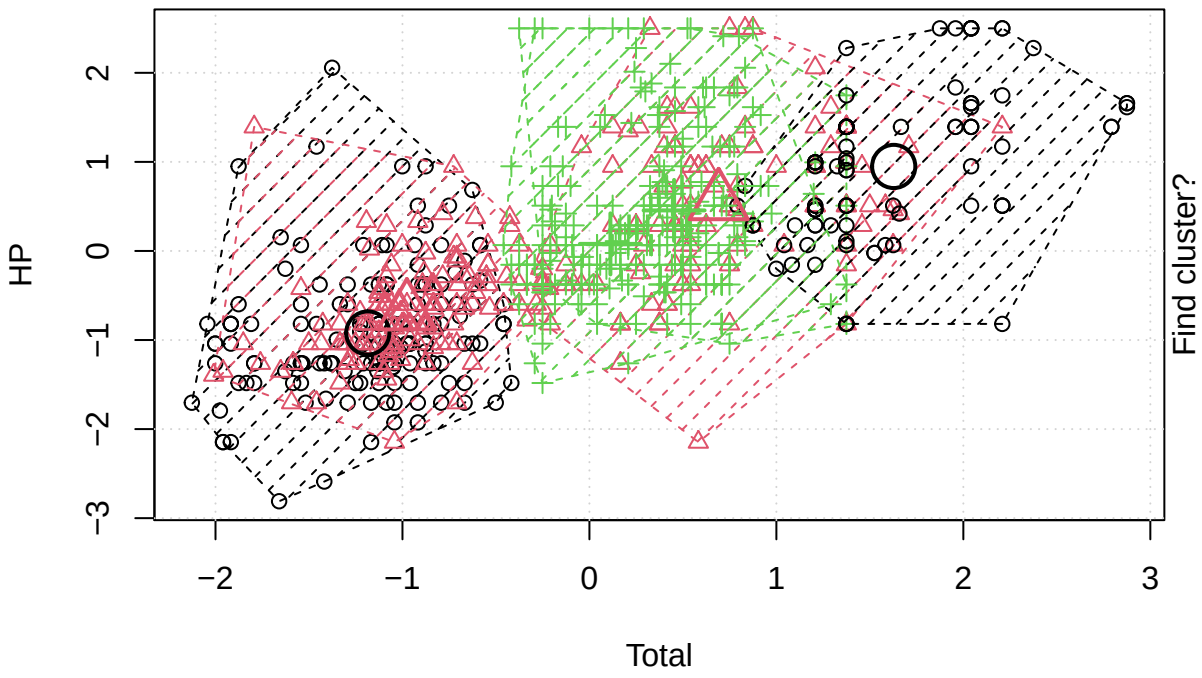

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

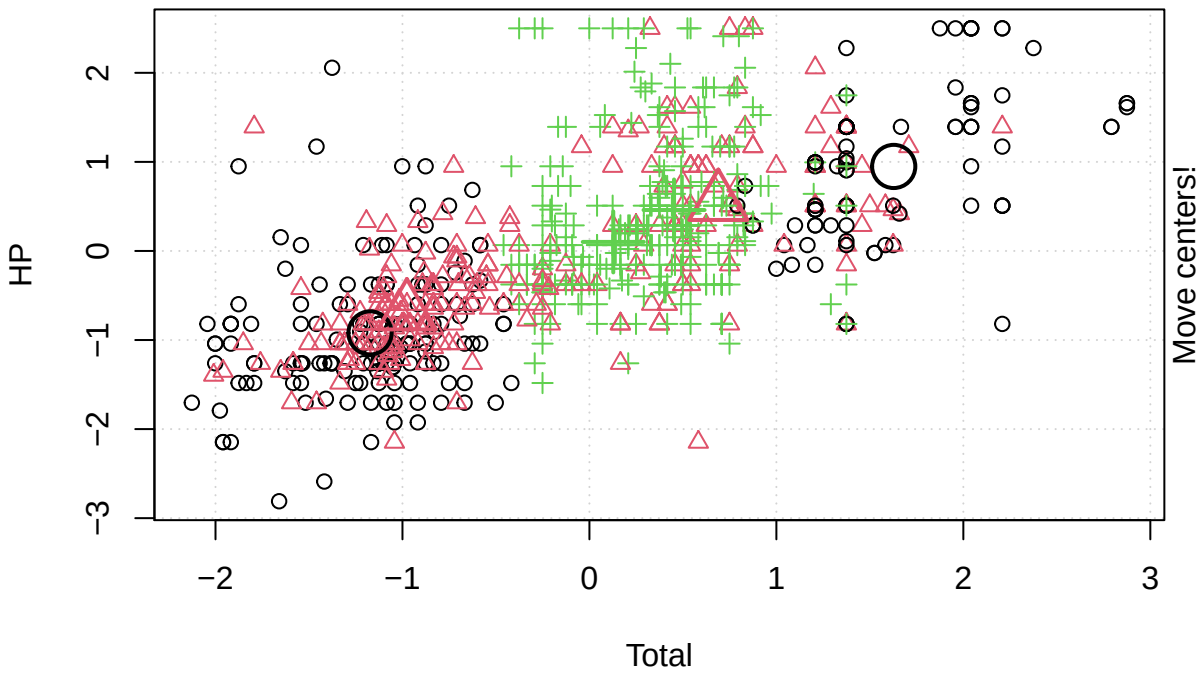

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

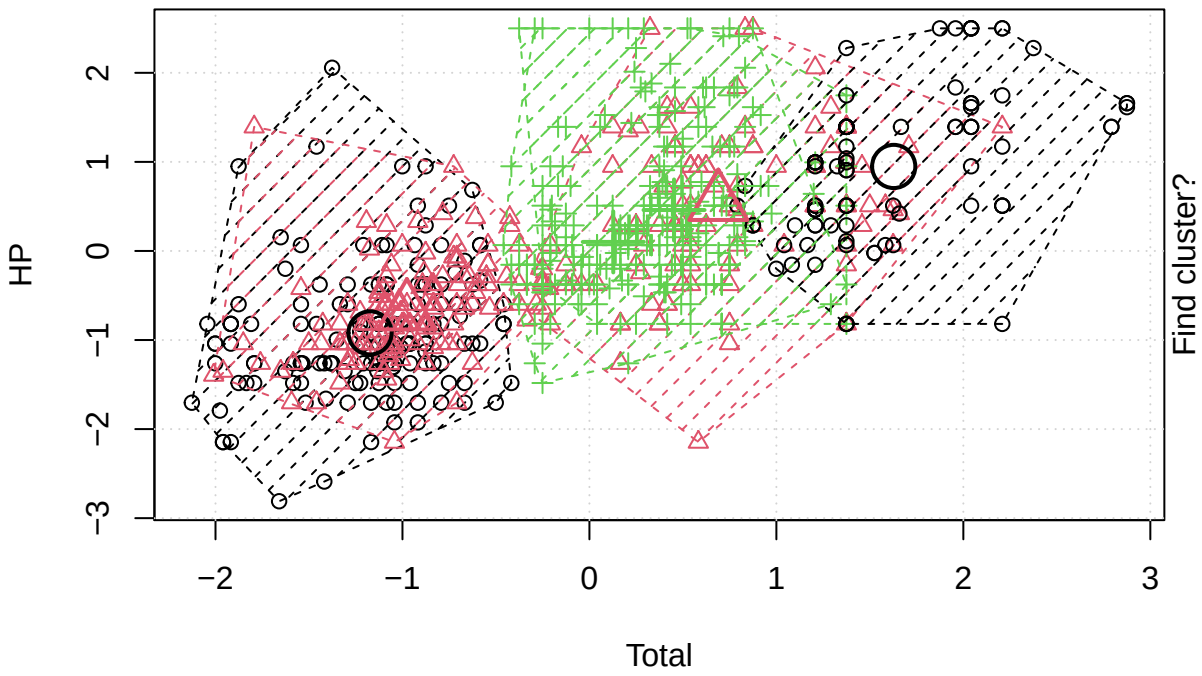

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

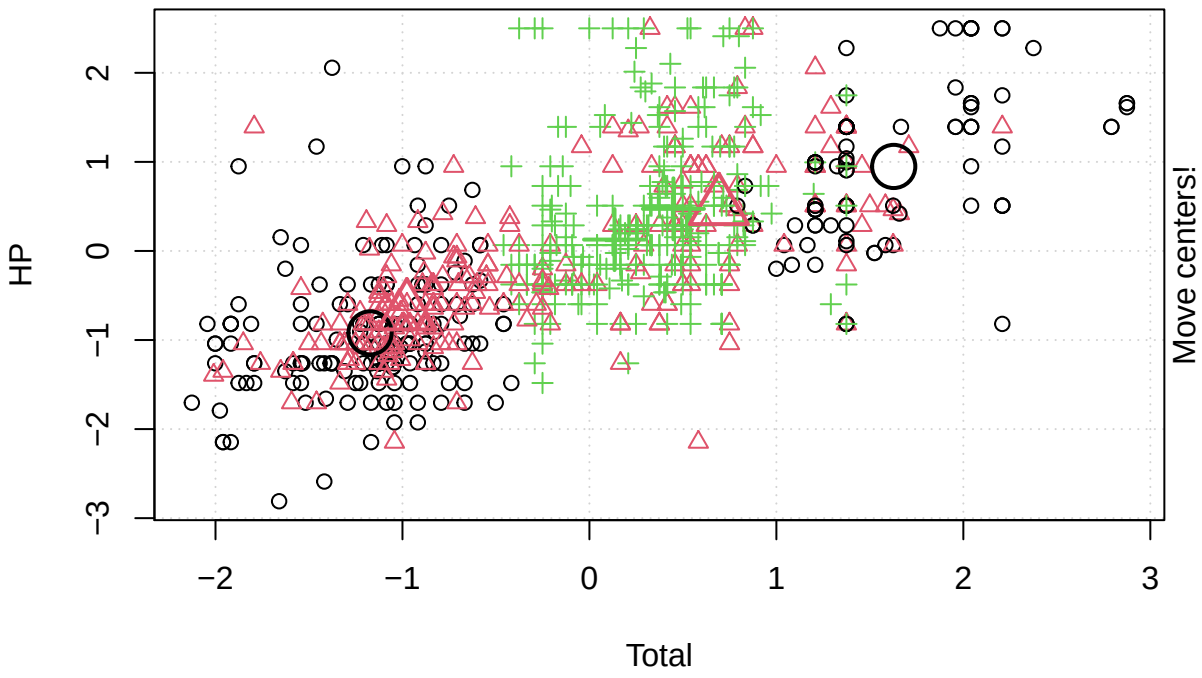

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

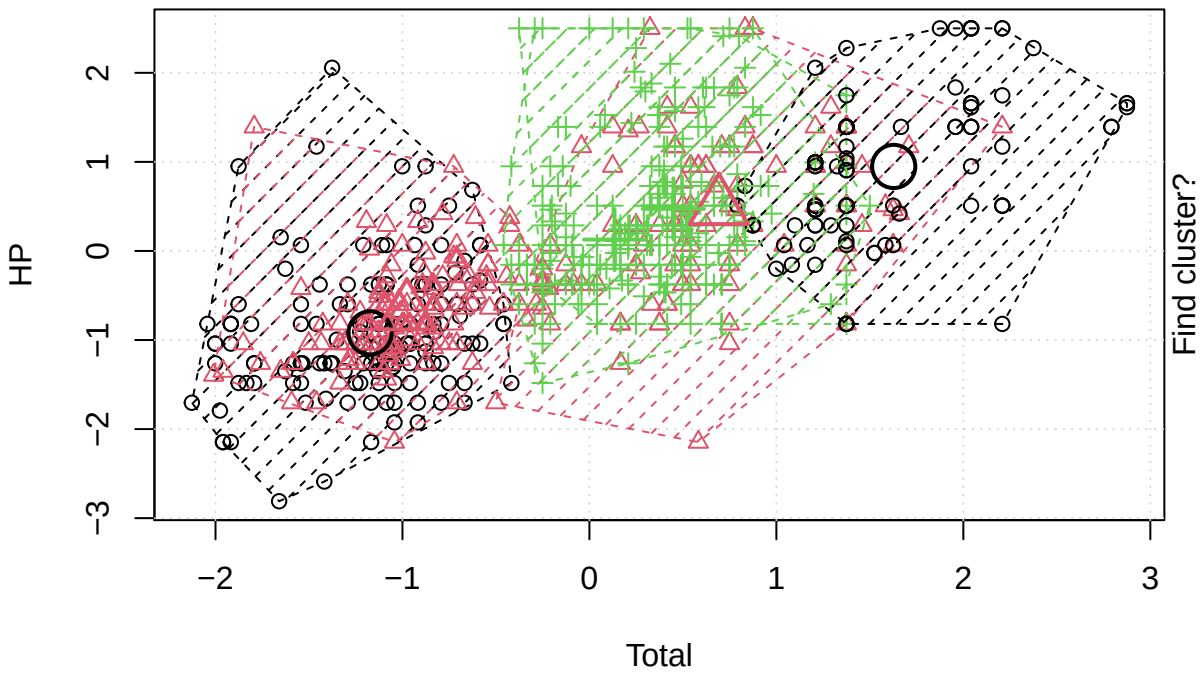

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

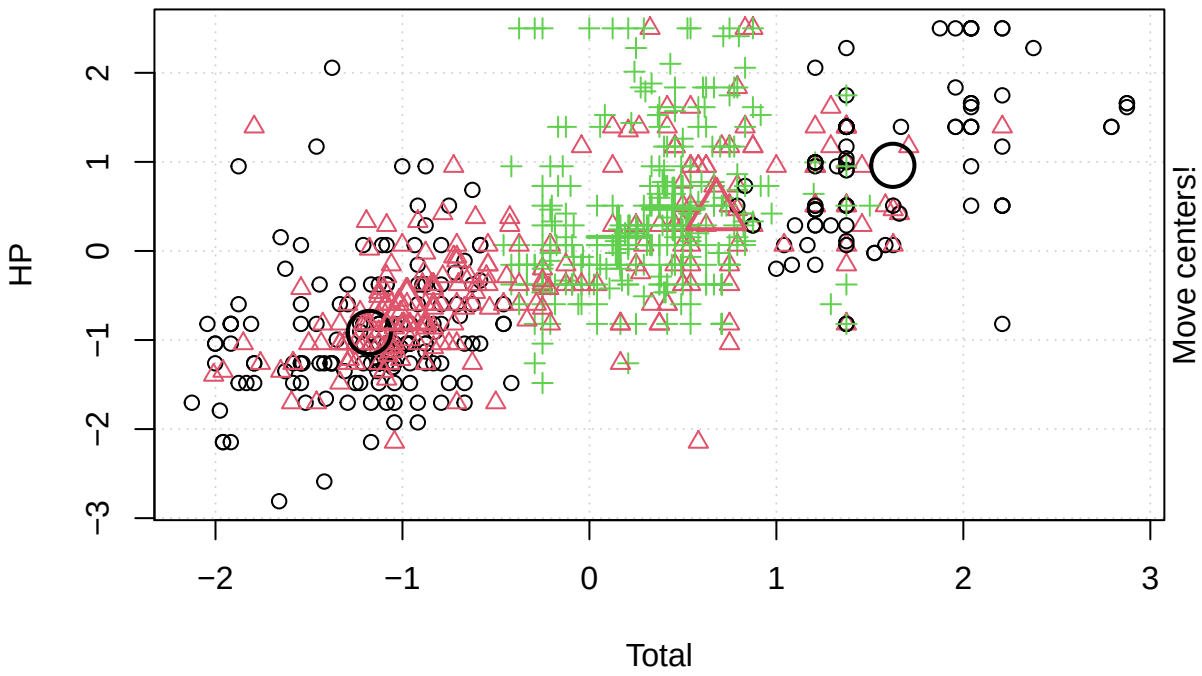

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

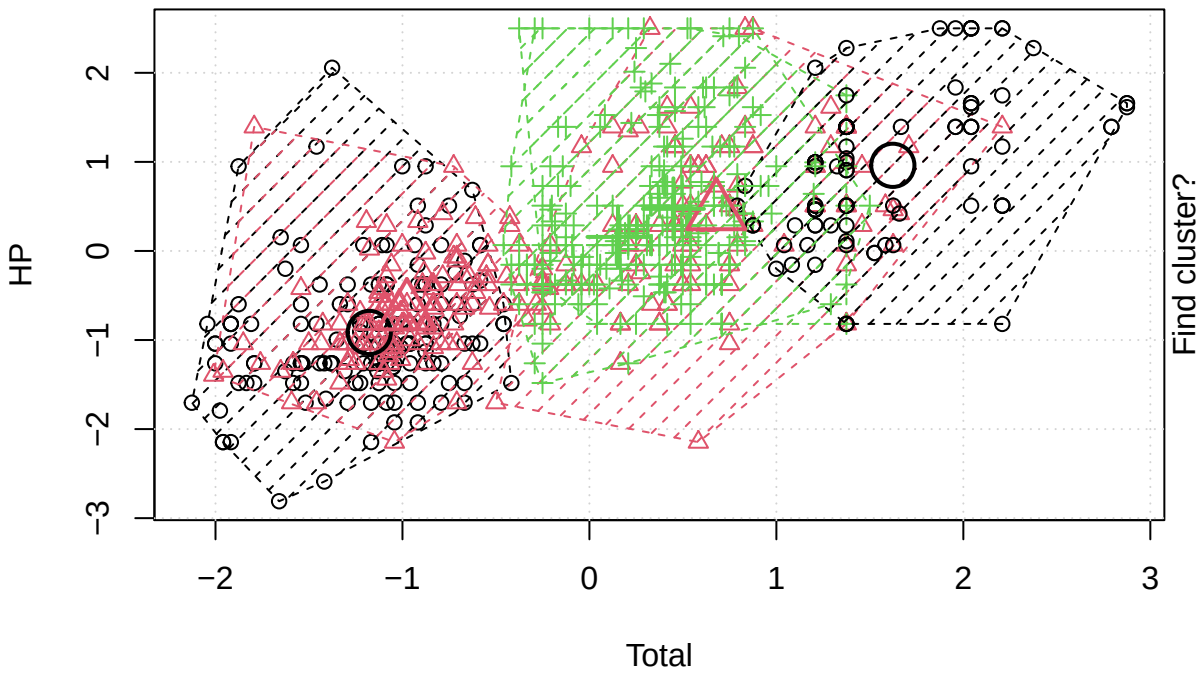

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

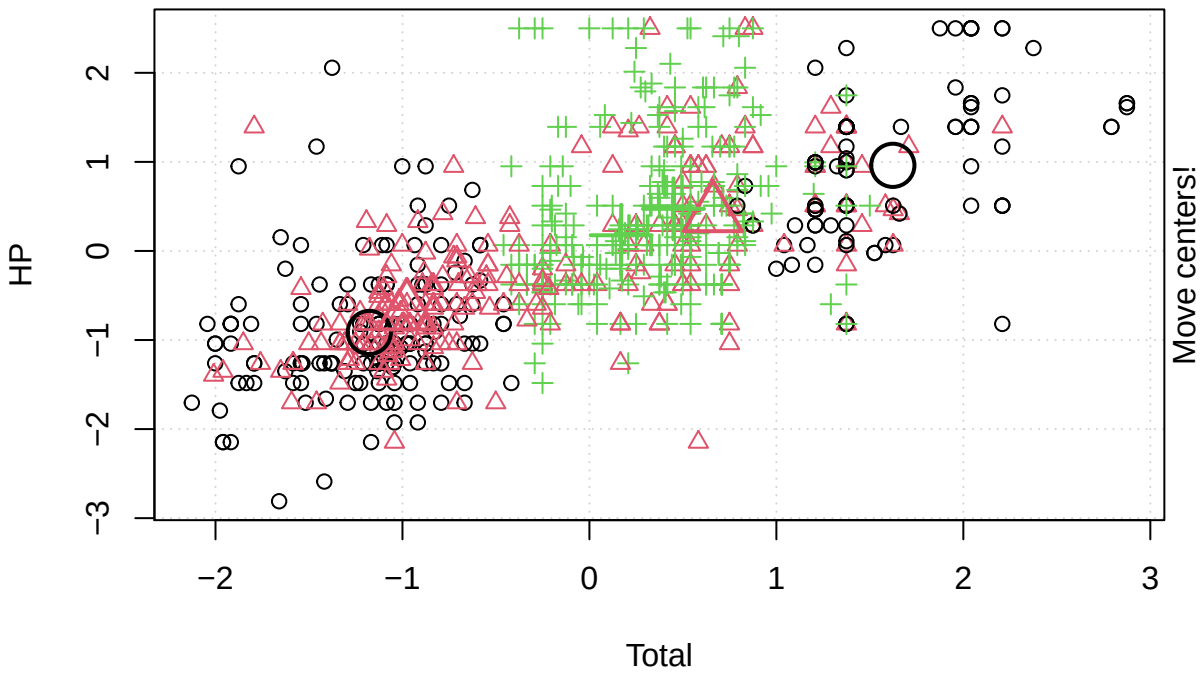

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

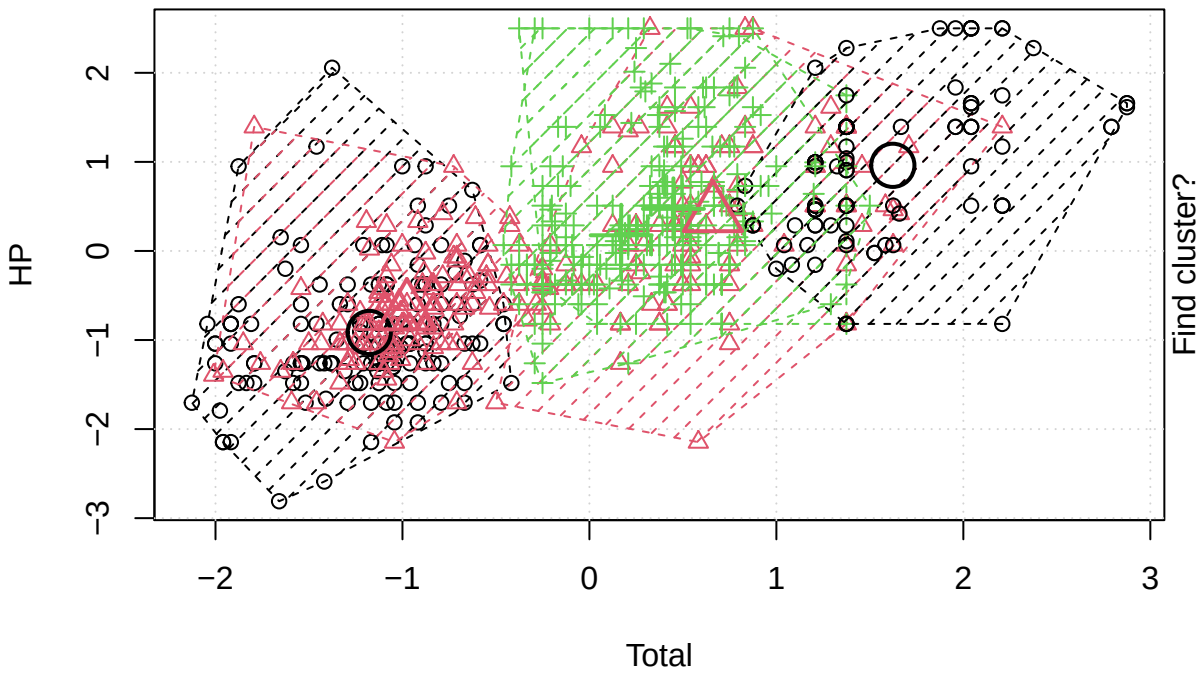

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

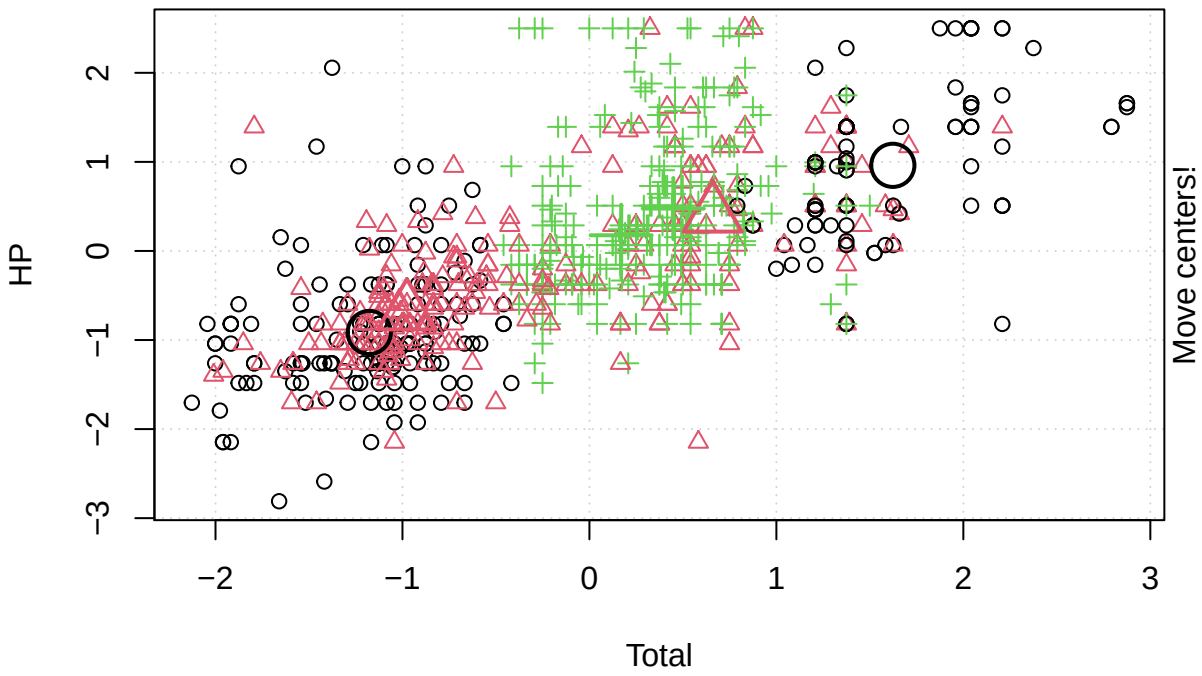

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

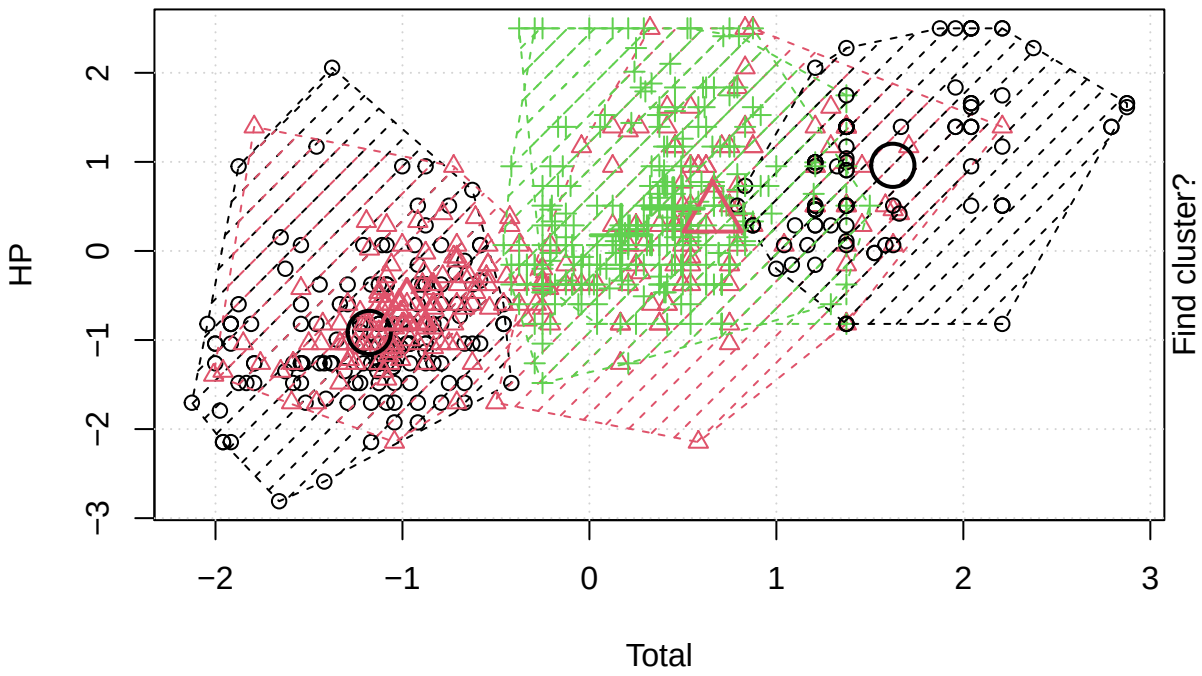

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

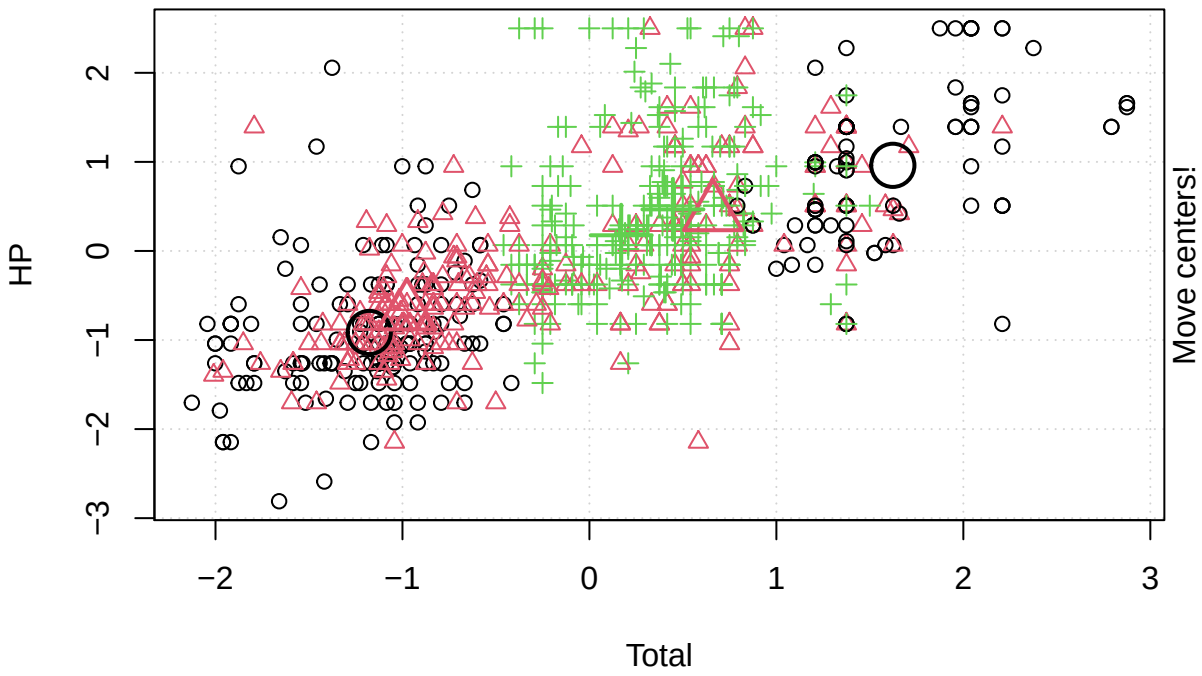

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

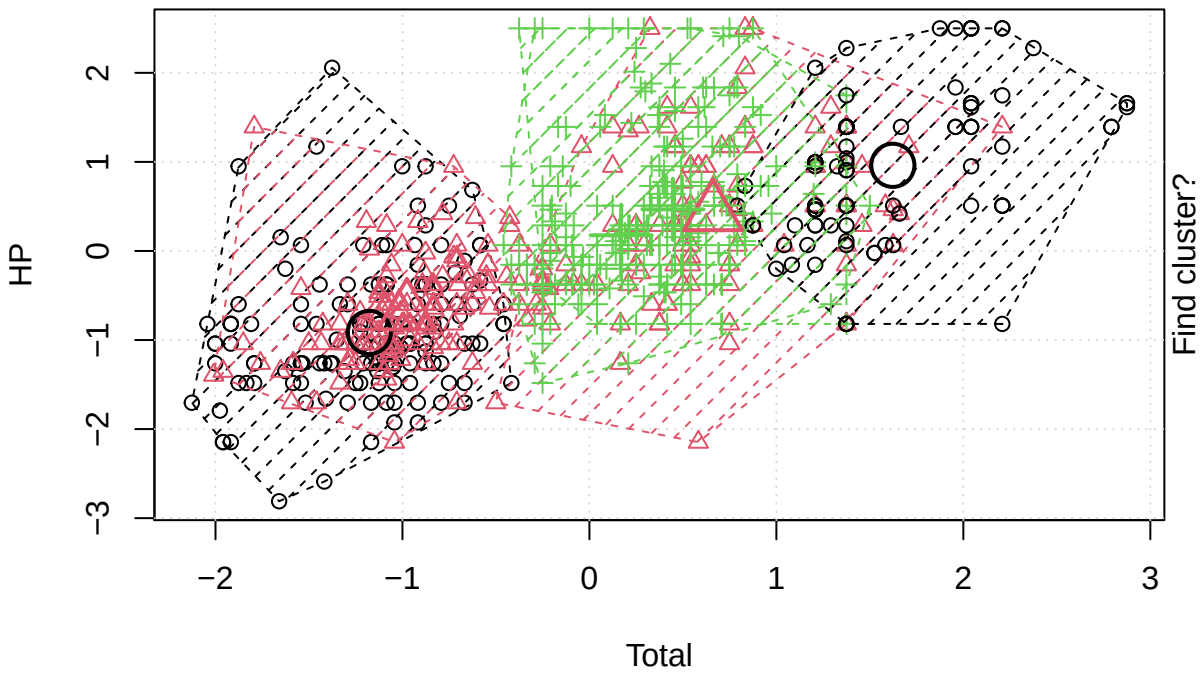

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

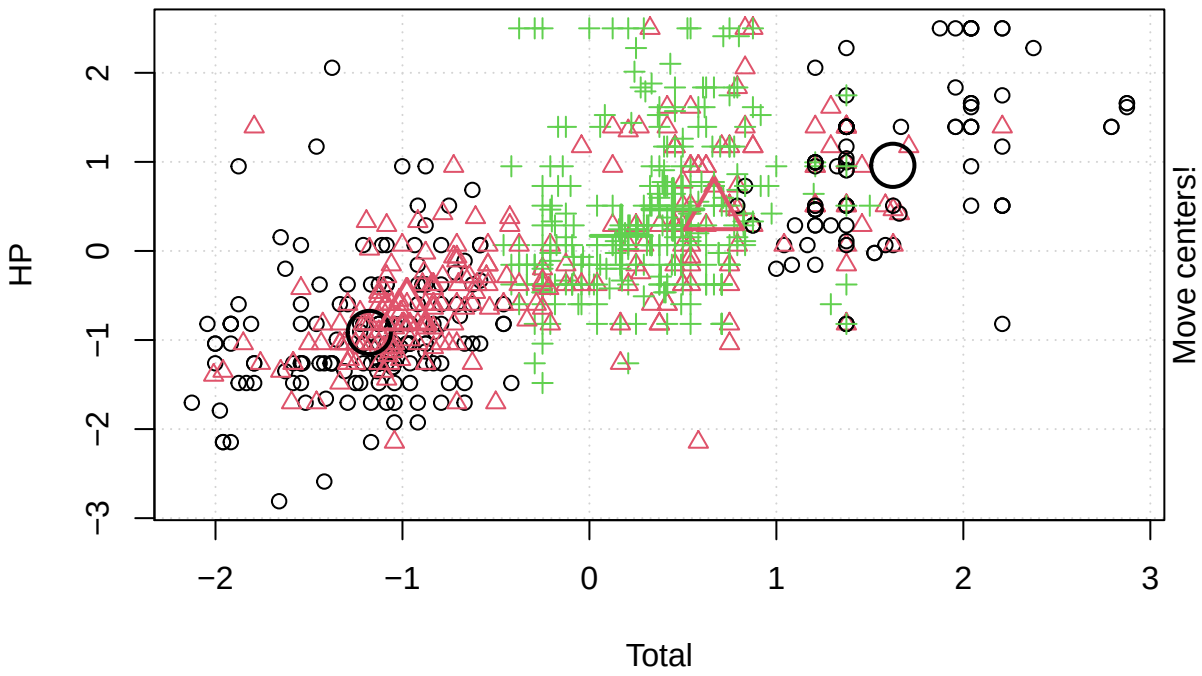

HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

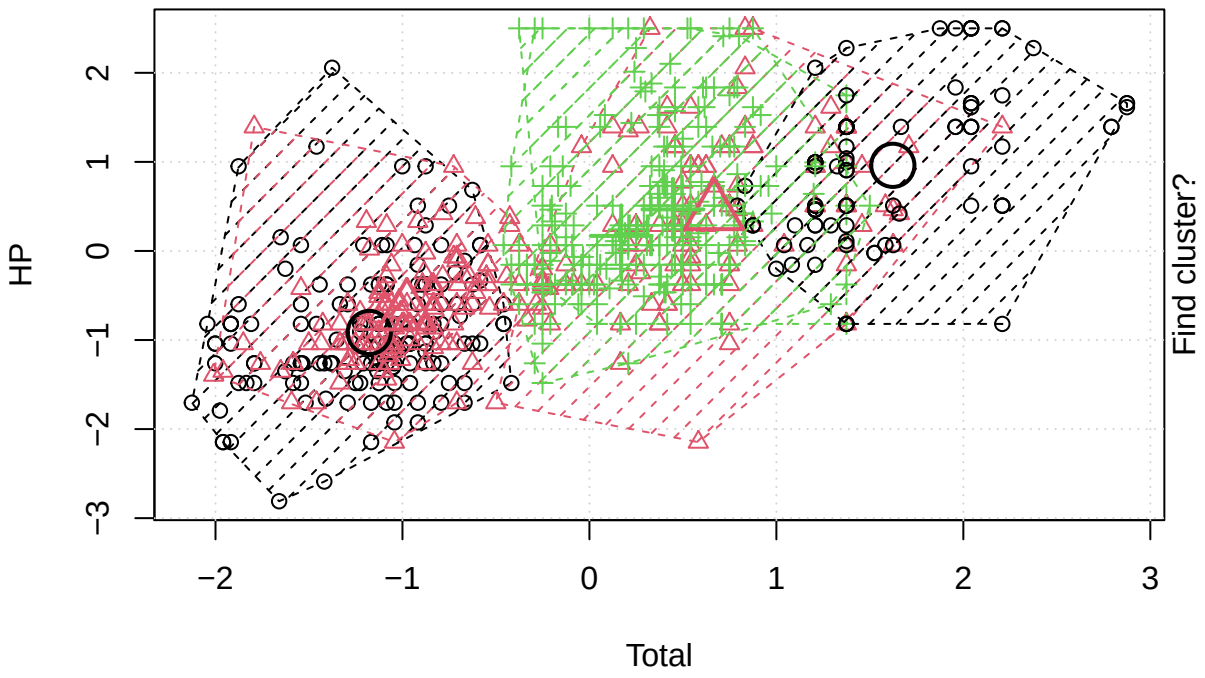

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3


HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3


HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3


HP
Total
Find cluster?
−3
−2
−1
0
1
2
−2
−1
0
1
2
3

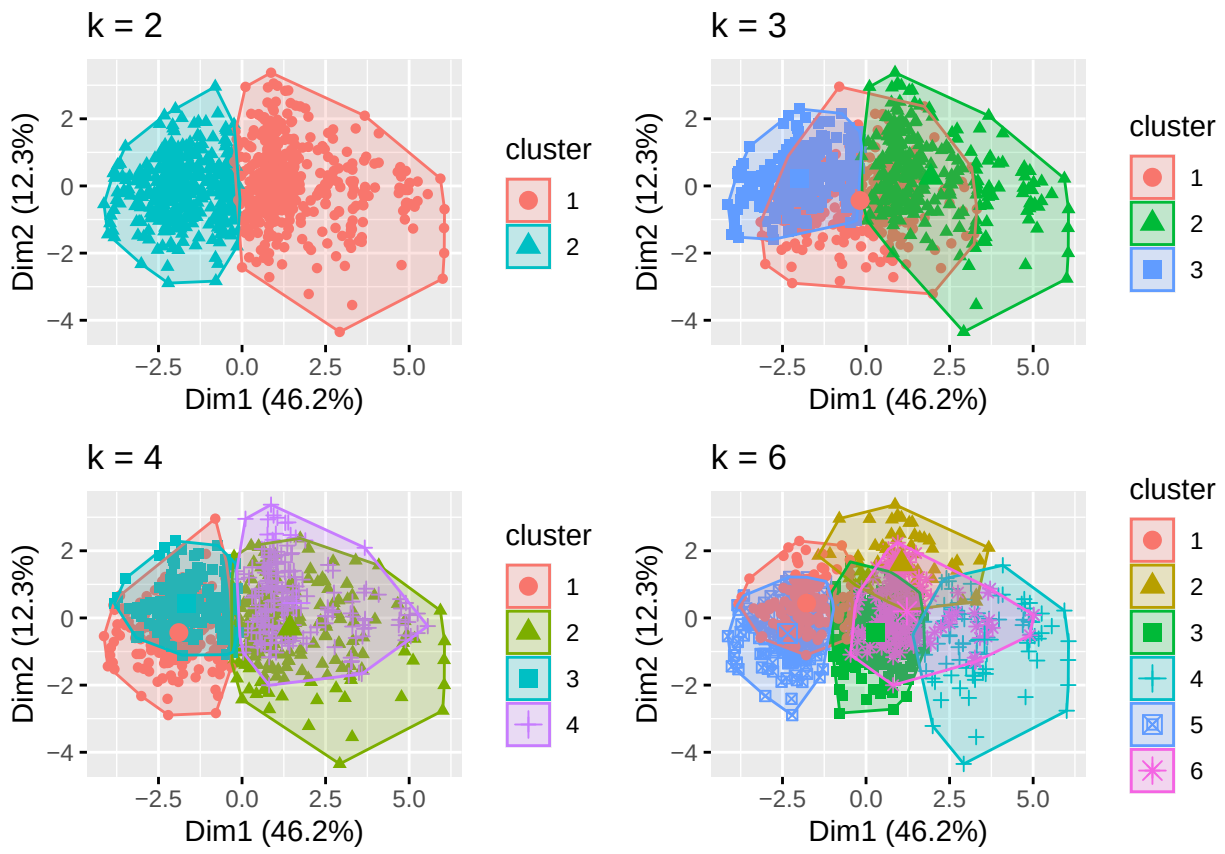

HP
Total
Move centers!
−3
−2
−1
0
1
2
−2
−1
0
1
2
3



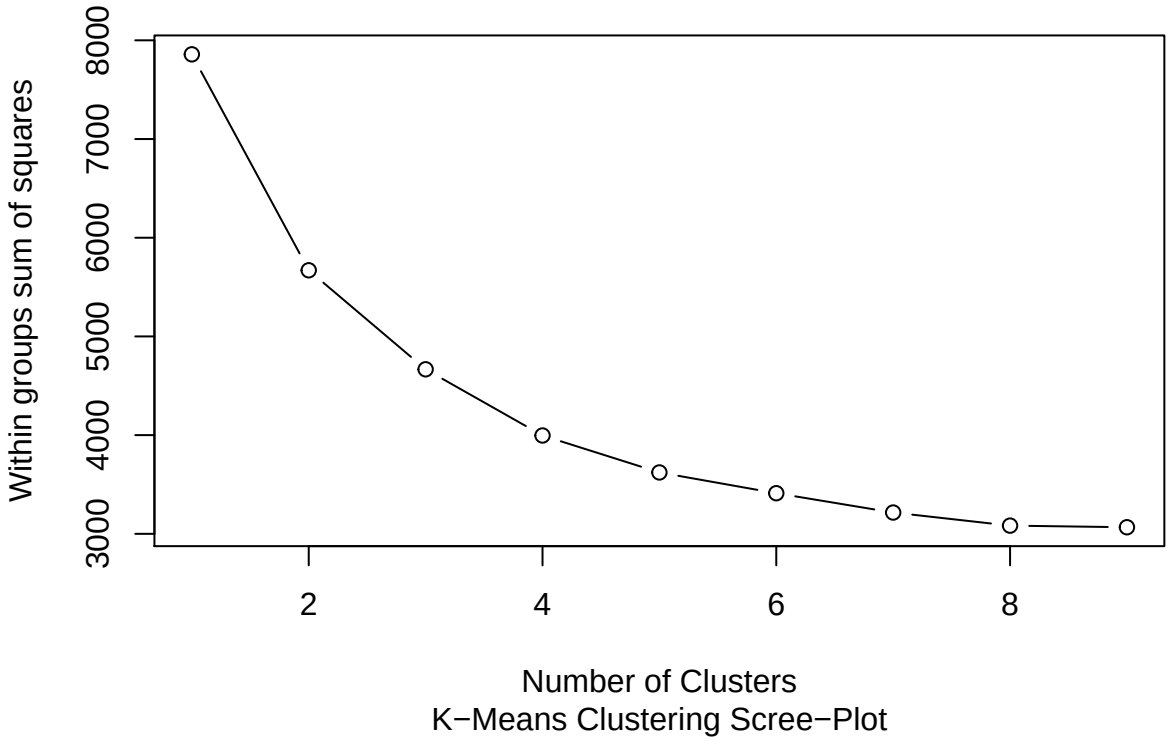

```
k2=kmeans(pokemonn,centers = 2,nstart = 25)
k3 <- kmeans(pokemonn, centers = 3, nstart = 25)
k4 <- kmeans(pokemonn, centers = 4, nstart = 25)
k5 <- kmeans(pokemonn, centers = 6, nstart = 25)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

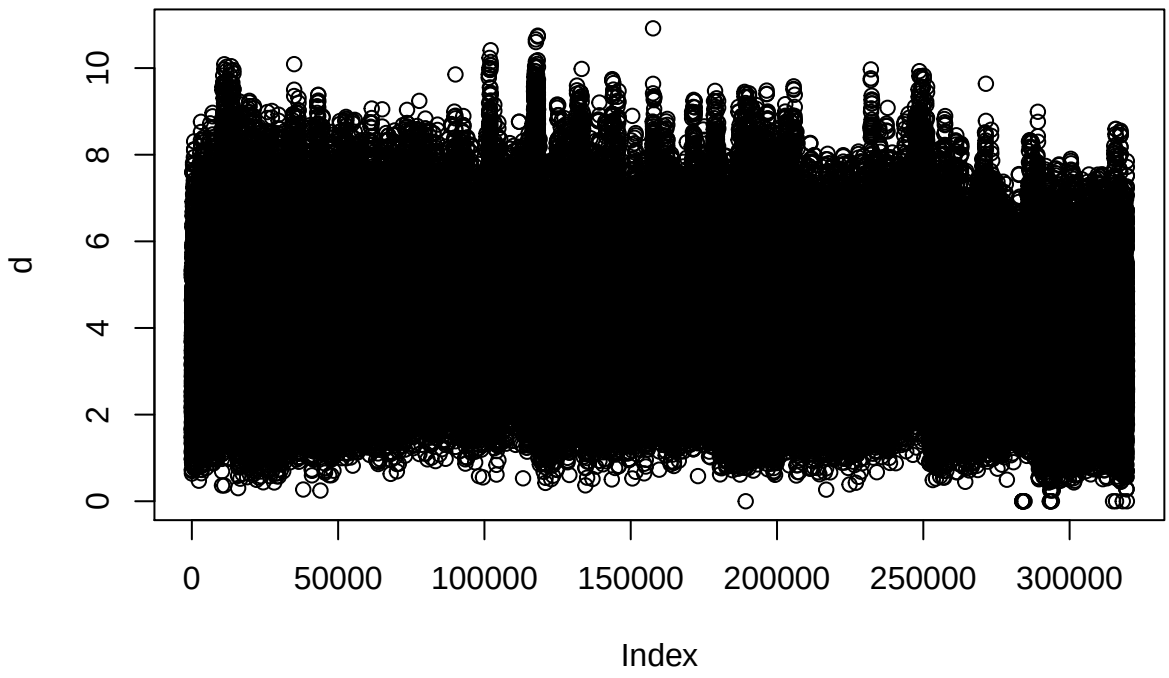
```
# plots to compare
p1 <- fviz_cluster(k2, geom = "point", data = pokemonn) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = pokemonn) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",  data = pokemonn) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",  data = pokemonn) + ggtitle("k = 6")

library(gridExtra)
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



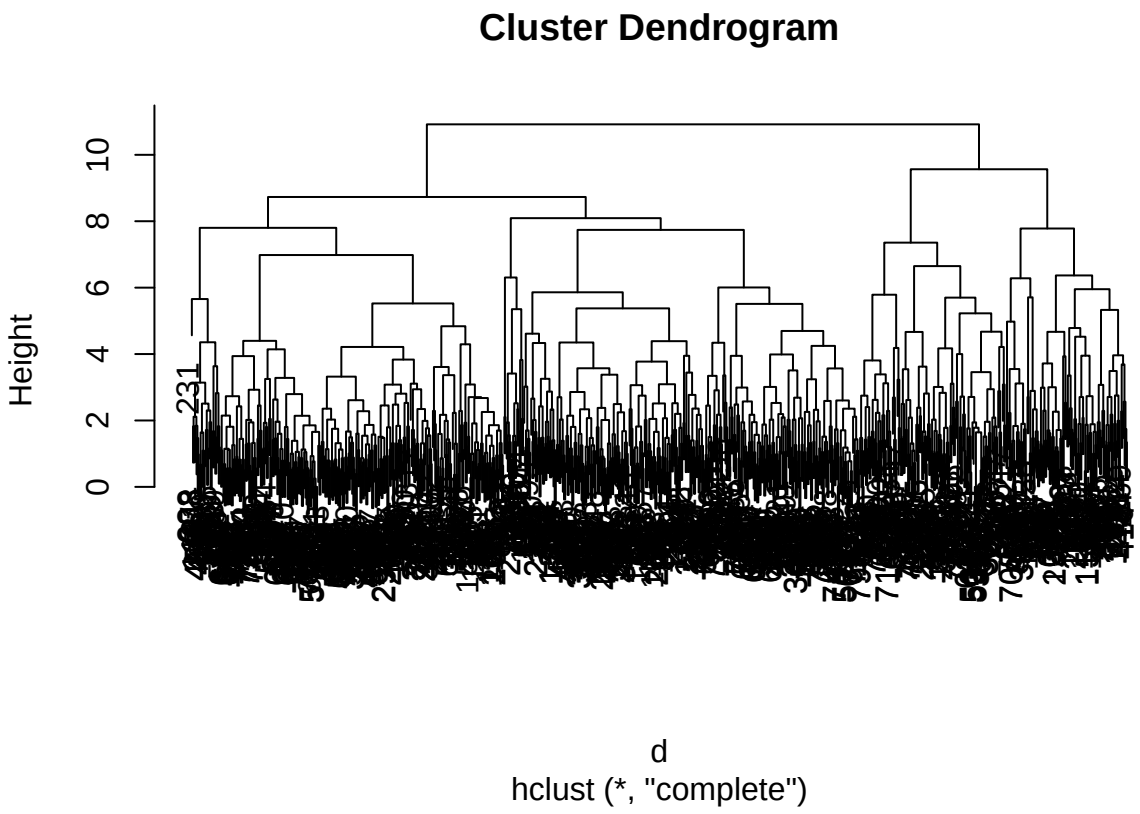

```
# TO decide the k valus use elbow curve/ scree plot or user  k ~ sqrt(n/2)
# Determine number of clusters by scree-plot
wss = (nrow(pokemonn))*sum(apply(pokemonn,2, var))
for (i in 1:9) wss[i] = sum(kmeans(pokemonn, centers=i)$withinss)
plot(1:9, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares") # Look for an
title(sub ="K-Means Clustering Scree-Plot")
```




```
################## 2.h-clustering ###############

d<-dist(pokemonn,method="euclidean")
plot(d)
```




```
#hclustering model
fit<-hclust(d,method="complete")
#check the clustering using dendogram
plot(fit)
```

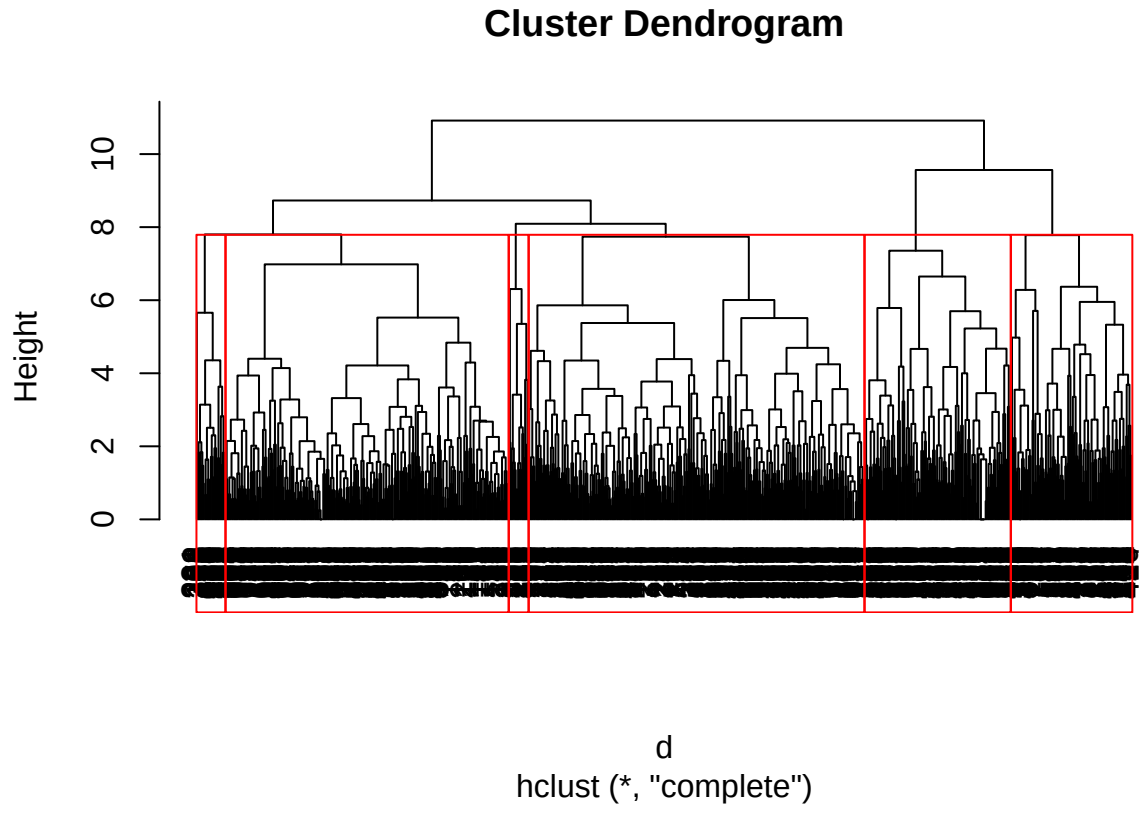



```
#check all variables in one line
plot(fit,hang=-1)
#looking in to dendogram decide how many clusters we need
#we are dividing dendogramin to 6 clusters by taking k=6
cutree(fit,k=6)
```

```
##   [1] 1 1 2 3 1 2 3 3 3 1 1 2 3 1 1 2 1 1 2 3 1 2 2 3 1 2 1 2 1 2 1 2 2 2 1 2 2
##  [38] 1 2 2 1 3 1 2 4 4 1 2 1 1 3 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 1 3 3 4 2
##  [75] 2 1 1 2 2 2 2 2 2 2 2 4 3 3 1 2 1 1 2 1 2 4 3 2 2 1 1 3 3 2 1 2 2 2 1 3 1
## [112] 3 1 2 2 2 4 1 2 4 2 4 2 2 3 1 2 1 2 1 3 2 2 3 2 2 2 3 2 1 3 3 3 1 1 3 3 2
## [149] 1 1 2 2 2 2 3 3 3 3 3 1 2 3 3 3 3 3 1 5 6 2 2 3 1 2 6 1 2 1 3 1 2 1 2 2 2
## [186] 3 1 1 4 1 5 2 2 1 2 3 3 3 1 4 2 3 1 2 2 2 1 3 2 1 4 3 6 2 3 2 2 4 2 1 6 4
## [223] 2 6 6 1 2 2 3 3 5 2 3 2 1 2 1 5 1 2 5 2 3 2 2 2 2 2 2 3 6 4 2 3 2 1 1 2 1 2
## [260] 2 2 4 3 3 3 1 2 3 3 3 3 3 1 2 2 3 1 2 2 3 1 2 2 3 1 2 1 2 1 1 2 1 2 1 1 6
## [297] 1 2 2 1 2 1 5 1 1 6 3 1 2 1 2 1 2 3 1 2 1 1 2 2 1 2 1 5 1 2 1 6 1 5 1 6 6
## [334] 6 1 2 6 1 2 3 2 2 2 2 2 1 2 2 2 3 2 2 1 3 3 6 1 6 2 1 2 6 1 2 1 5 3 2 2 2
## [371] 2 1 2 1 2 1 5 5 3 2 3 1 6 2 5 1 2 3 5 6 2 2 2 3 4 1 6 3 1 2 2 1 2 2 6 1 1
## [408] 2 3 3 1 5 3 3 6 6 6 6 3 6 3 3 3 3 3 3 3 6 3 3 6 3 1 1 6 1 2 2 1 2 2 1 1 2
## [445] 1 2 1 2 1 2 2 1 6 1 2 5 6 1 5 5 5 2 1 5 2 1 2 1 2 1 2 2 2 2 1 2 2 6 2 1 2
## [482] 1 1 2 5 5 1 5 4 2 5 1 2 6 3 2 1 2 3 1 6 1 2 1 2 2 1 2 5 1 2 3 2 6 2 6 6 2
## [519] 2 6 2 2 6 2 2 2 2 2 6 6 2 6 6 6 6 6 6 6 6 3 6 6 6 3 6 6 6 2 6 3 6 3 6 6 1
## [556] 6 2 1 2 2 1 2 2 1 6 1 1 2 1 2 1 2 1 2 1 2 1 2 1 1 2 1 2 1 1 6 1 2 1 2 2 6
## [593] 1 1 2 1 2 2 2 2 1 1 2 1 1 2 1 2 1 6 2 1 1 2 1 2 6 2 1 6 1 6 6 1 6 1 6 2 6
## [630] 1 2 1 6 1 2 1 6 6 1 1 2 1 2 1 6 6 1 2 2 1 6 1 2 1 2 2 1 2 5 6 1 6 2 1 6 2
## [667] 1 6 1 1 6 1 2 2 1 2 6 1 2 2 1 6 2 1 2 1 2 2 1 2 1 2 2 2 1 6 6 1 6 6 6 6 3
## [704] 3 3 3 6 3 6 6 6 3 6 6 6 6 6 6 1 1 6 1 2 2 1 2 2 1 2 1 2 2 1 1 2 2 2 1 1 6
```

```
## [741] 1 2 1 2 2 2 2 2 1 6 6 6 1 2 1 2 1 2 1 6 1 6 1 6 1 2 1 6 1 2 6 2 2 6 1 6 6
## [778] 6 1 2 1 1 1 1 2 2 6 6 1 6 1 2 6 6 6 6 6 6 6 6
```

```
# labelling the groups with red colour
rect.hclust(fit,k=6,border="red")
```




```
View(pokemonn)

############## OBSERVATIONS #############
# 1.1st,3rd,5th generations pokemons is more compare to other generations but only
#  3rd generation pokemons more Legendarys.
# 2.we have to group by calculating centrioid and finding the distance between
# centroids and variable and group the variable as per least distance we have taken six clusters
```