Untitled

```
diabetes<-read.csv("C:/Users/Kun Hu/Documents/pima-indians-diabetes.txt",header = F)
summary(diabetes)
```

```
##        V1               V2              V3               V4
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##        V5              V6              V7               V8
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725   Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##        V9
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
str(diabetes)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ V1: int  6 1 8 1 0 5 3 10 2 8 ...
##  $ V2: int  148 85 183 89 137 116 78 115 197 125 ...
##  $ V3: int  72 66 64 66 40 74 50 0 70 96 ...
##  $ V4: int  35 29 0 23 35 0 32 0 45 0 ...
##  $ V5: int  0 0 0 94 168 0 88 0 543 0 ...
##  $ V6: num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ V7: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ V8: int  50 31 32 21 33 30 26 29 53 54 ...
##  $ V9: int  1 0 1 0 1 0 1 0 1 1 ...
```

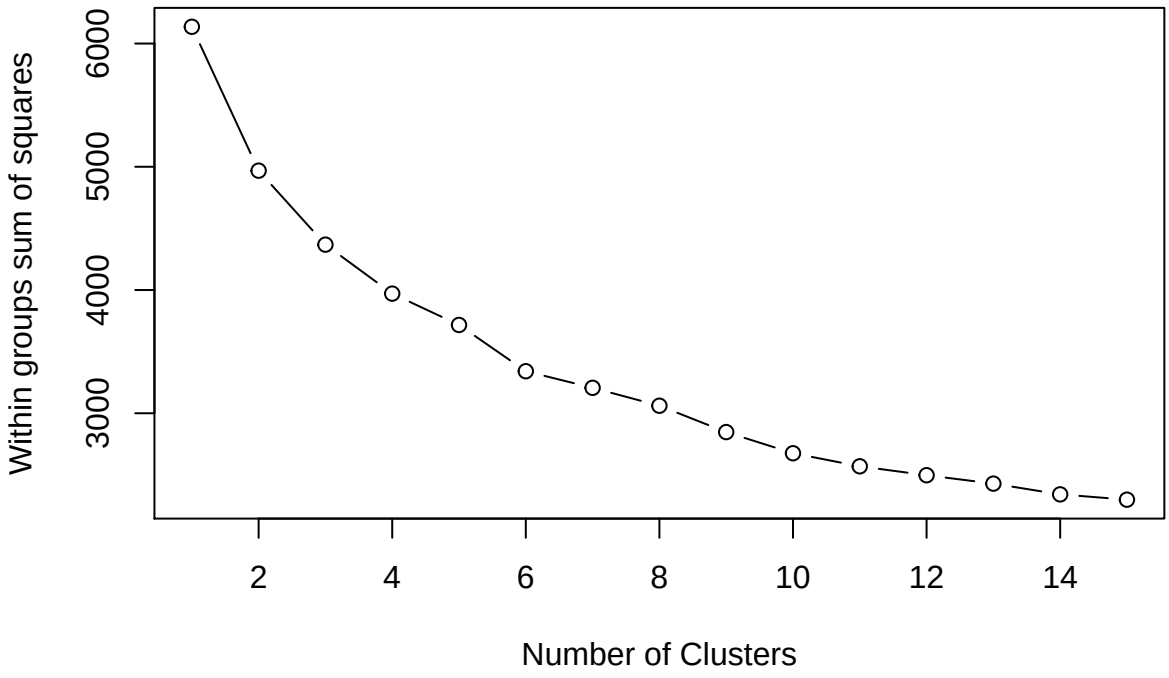
```
df<-scale(diabetes[-1])
str(df)
```

```
##  num [1:768, 1:8] 0.848 -1.123 1.942 -0.998 0.504 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:8] "V2" "V3" "V4" "V5" ...
##  - attr(*, "scaled:center")= Named num [1:8] 120.9 69.1 20.5 79.8 32 ...
##   ..- attr(*, "names")= chr [1:8] "V2" "V3" "V4" "V5" ...
##  - attr(*, "scaled:scale")= Named num [1:8] 31.97 19.36 15.95 115.24 7.88 ...
##   ..- attr(*, "names")= chr [1:8] "V2" "V3" "V4" "V5" ...
```

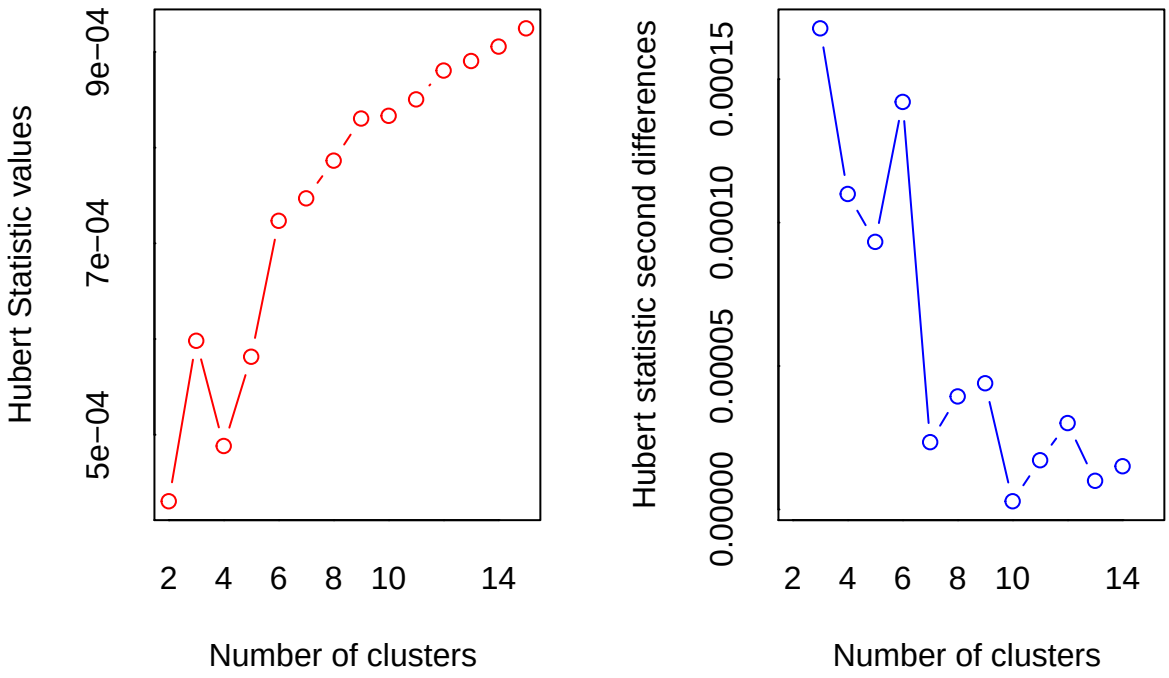
```
wssplot <- function(data, nc=15, seed=1234){
wss <- (nrow(data)-1)*sum(apply(data,2,var))
for (i in 2:nc){
```

```
set.seed(seed)
wss[i] <- sum(kmeans(data, centers=i)$withinss)}
plot(1:nc, wss, type="b", xlab="Number of Clusters",
ylab="Within groups sum of squares")
return(wss)}
```

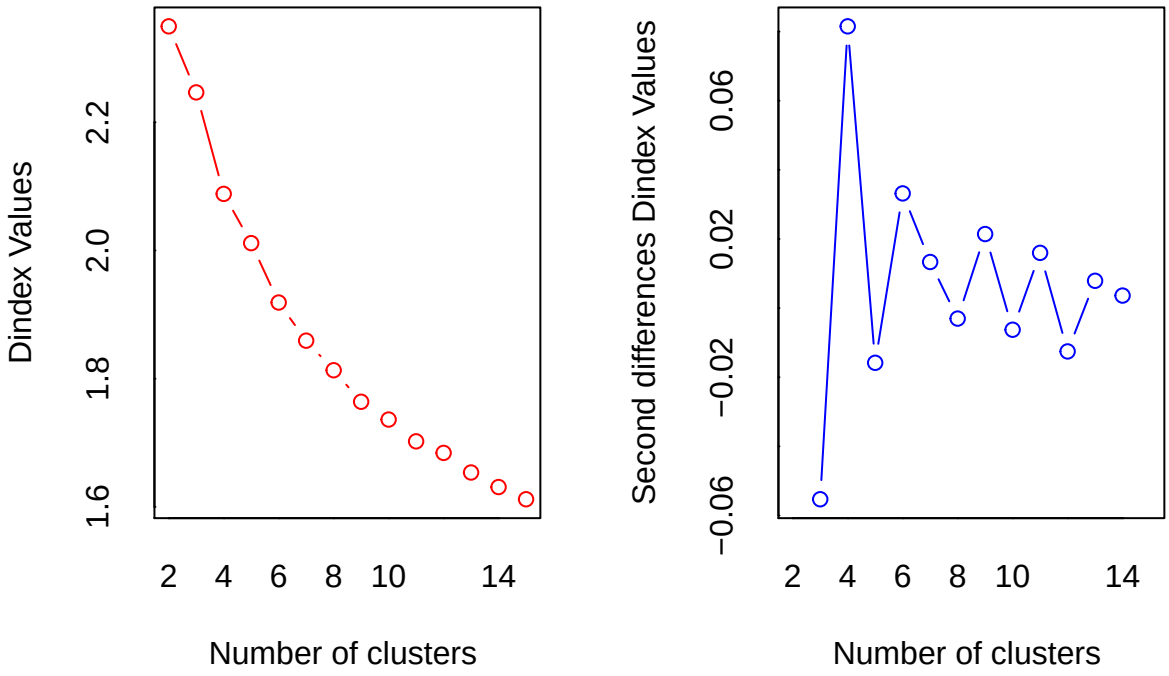
```
nc1<-wssplot(df)
```




```
library(NbClust)
nc2<-NbClust(df,min.nc=2,max.nc=15,method="kmeans")
```




```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 6 proposed 2 as the best number of clusters
## * 6 proposed 3 as the best number of clusters
## * 4 proposed 4 as the best number of clusters
## * 3 proposed 6 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 1 proposed 12 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
## * 1 proposed 15 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

```
nc2$Best.nc
```

```
##                      KL       CH Hartigan    CCC    Scott      Marriot
## Number_clusters 2.0000   2.0000   6.0000 15.000   3.0000 3.000000e+00
```

```
## Value_Index     8.7533 180.0254  38.1269 10.762 970.9509 2.393482e+22
##                    TrCovW   TraceW Friedman   Rubin Cindex      DB
## Number_clusters      4.0   4.0000   7.0000  6.0000 6.0000 12.0000
## Value_Index     104927.8 207.1382   3.0486 -0.0707 0.2256  1.5327
##                 Silhouette   Duda PseudoT2  Beale Ratkowsky     Ball
## Number_clusters       3.00 2.0000   2.0000 2.0000    4.0000   3.0000
## Value_Index           0.24 0.8881  62.5033 0.6646    0.2893 994.2887
##                 PtBiserial Frey McClain  Dunn Hubert SDindex Dindex
## Number_clusters     3.0000    1  2.0000 3.000      0  4.0000      0
## Value_Index         0.4846   NA  0.6611 0.073      0  1.9556      0
##                     SDbw
## Number_clusters 14.0000
## Value_Index      0.5185
```

```
set.seed(6395)
responseY<-diabetes[,9]
predictorX<-df[,1:8]
pca<-princomp(predictorX,cor=T)
pca$sdev
```

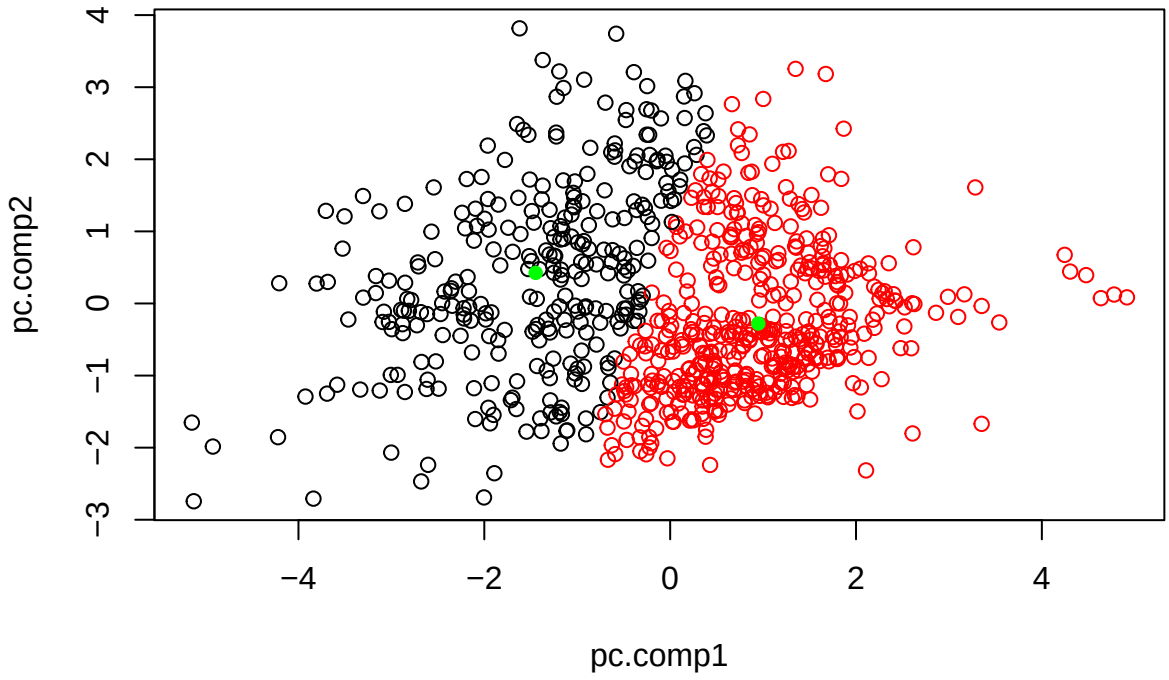
```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 1.5172950 1.1975654 1.0453350 0.9350144 0.9084949 0.7705277 0.6976597
##    Comp.8
## 0.6251967
```

```
summary(pca)
```

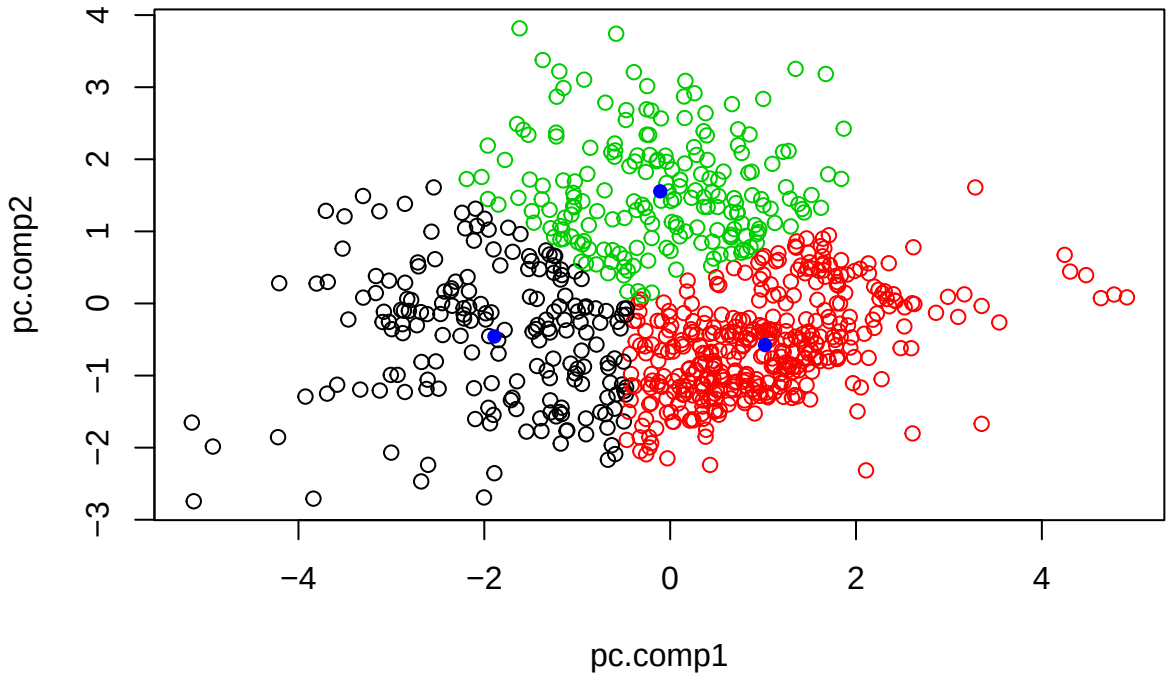
```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4    Comp.5
## Standard deviation     1.517295 1.1975654 1.0453350 0.9350144 0.9084949
## Proportion of Variance 0.287773 0.1792703 0.1365907 0.1092815 0.1031704
## Cumulative Proportion  0.287773 0.4670434 0.6036340 0.7129155 0.8160859
##                           Comp.6     Comp.7     Comp.8
## Standard deviation     0.77052770 0.69765971 0.62519666
## Proportion of Variance 0.07421412 0.06084113 0.04885886
## Cumulative Proportion  0.89030001 0.95114114 1.00000000
```

```
pc.comp<-pca$scores
pc.comp1<-pc.comp[,1]
pc.comp2<-pc.comp[,2]
x<-cbind(pc.comp1,pc.comp2)
```

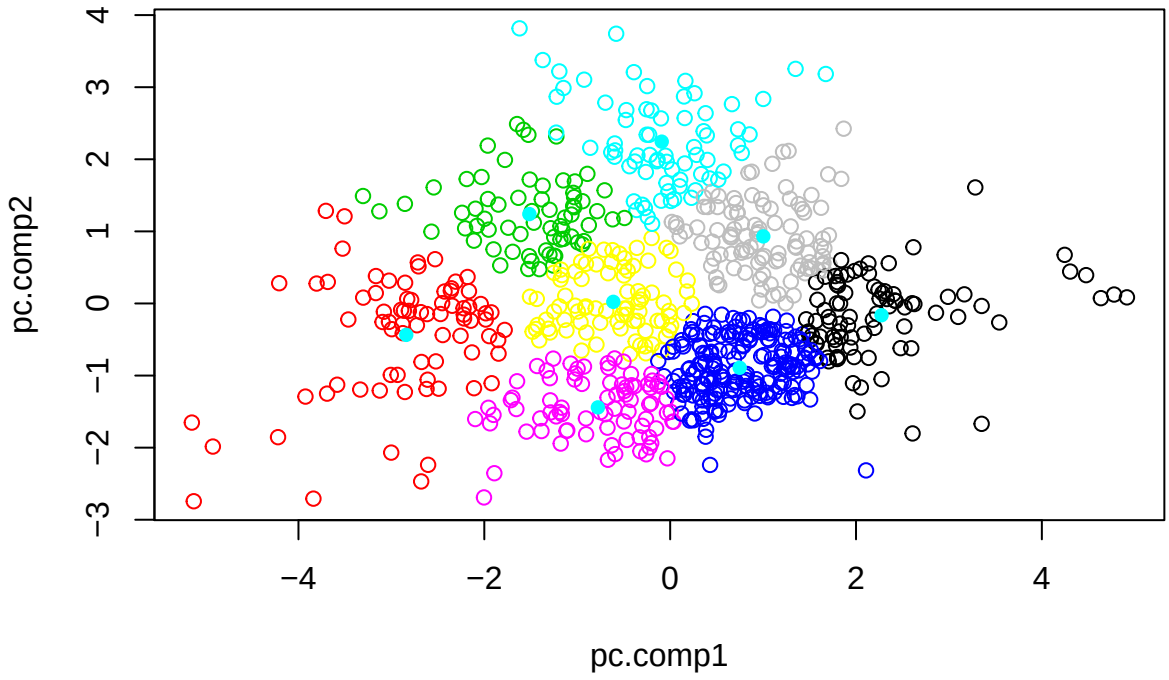
```
set.seed(1234)
cl<-kmeans(x,2)
plot(pc.comp1,pc.comp2,col=cl$cluster)
points(cl$centers,pch=16,col="green")
```



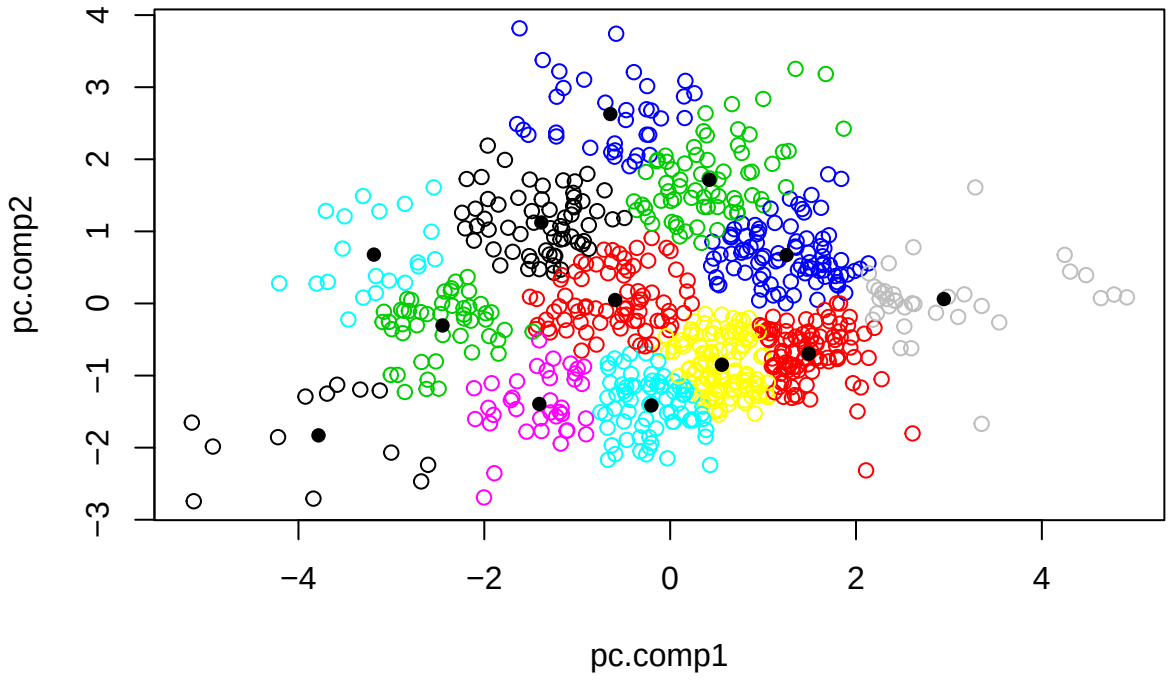

```
set.seed(2345)
cl<-kmeans(x,3)
plot(pc.comp1,pc.comp2,col=cl$cluster)
points(cl$centers,pch=16,col="blue")
```




```
set.seed(3456)
cl<-kmeans(x,8)
plot(pc.comp1,pc.comp2,col=cl$cluster)
points(cl$centers,pch=16,col="cyan")
```




```
set.seed(4567)
cl<-kmeans(x,13)
plot(pc.comp1,pc.comp2,col=cl$cluster)
points(cl$centers,pch=16)
```




```
cl$centers
```

```
##      pc.comp1    pc.comp2
## 1  -3.7845829 -1.82997218
## 2  -0.5919884  0.04779155
## 3   0.4213073  1.71606841
## 4   1.2519539  0.67032862
## 5  -3.1882220  0.67756772
## 6  -1.4084729 -1.39380205
## 7   0.5561967 -0.84983737
## 8   2.9454190  0.06105810
## 9  -1.3865936  1.12360068
## 10  1.4928372 -0.69496693
## 11 -2.4498710 -0.30526698
## 12 -0.6434765  2.62740683
## 13 -0.2027170 -1.41481864
```

```
table(diabetes$V9,cl$cluster)
```

```
##
##        1   2   3   4   5   6   7   8   9  10  11  12  13
##   0    5  36  35  72   1  25 112  34   7  98   1   4  70
##   1    8  41  34  15  19  12   3   1  55   0  47  31   2
```