Descriptive statistic and clustering

reminder about the descriptive statisic

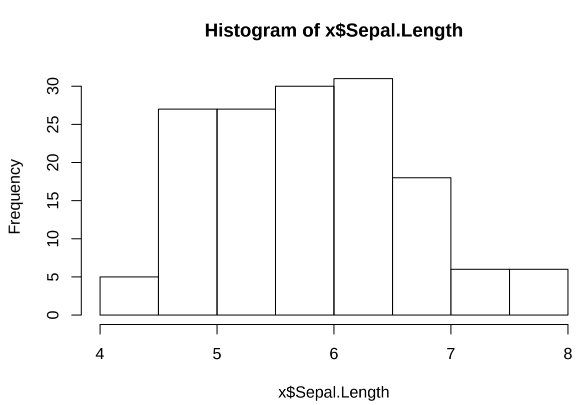
```
x=iris
summary(x)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
##                 
```

As we saw, descriptive statistics are useful to start discovering the data(here is obvious a supervise learning)

- about histogram: best choice by R (bins number)

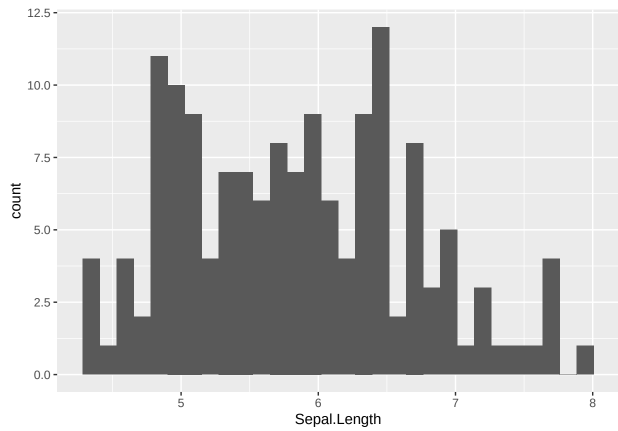
```
hist(x$Sepal.Length)
```



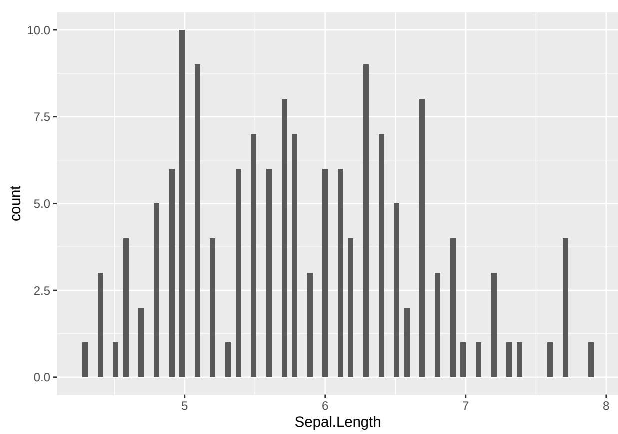
- best choice by ggplot (bins number)

```
library(ggplot2)
ggplot(x)+geom_histogram(aes(x=Sepal.Length))
```

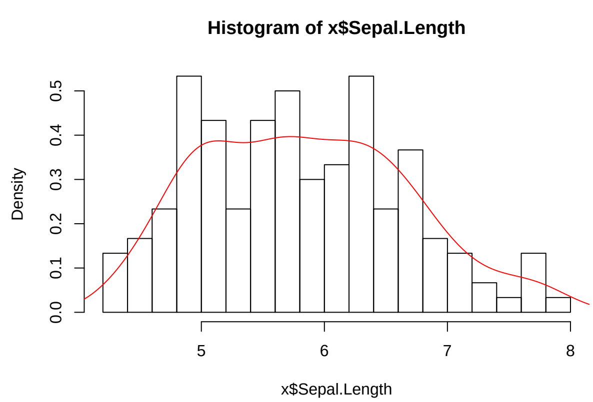
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
ggplot(x)+geom_histogram(aes(x=Sepal.Length),bins=100)
```

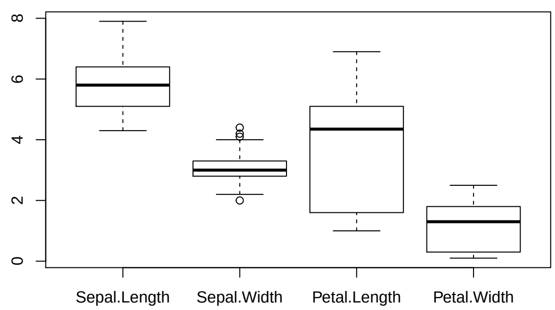



```r
hist(x$Sepal.Length,breaks=20,freq=FALSE)
lines(density(x$Sepal.Length),col='red')
```

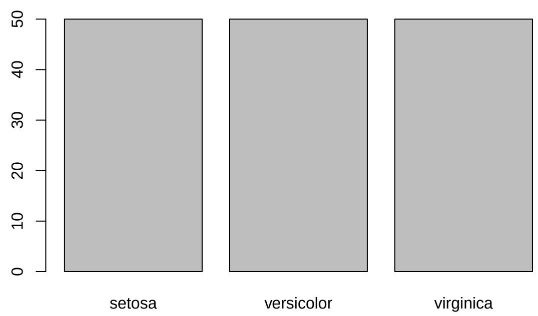


- boxplot check petal.length, here we have a distribution, very low to increase and very fast to decrease

```r
boxplot(x[,-5])
```

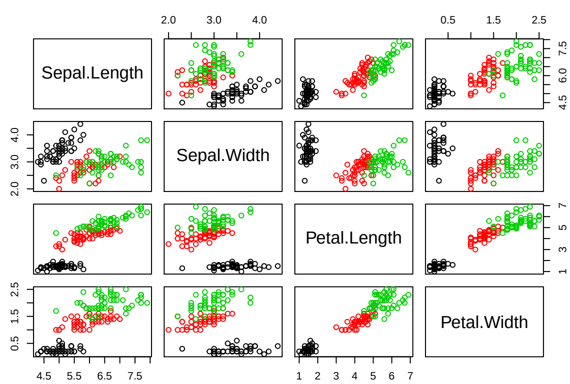



```r
barplot(summary(x$Species))
```



* try to pair to see the relationship

```r
pairs(x[,-5],col=as.numeric(x$Species))
```



Unsupervised learning: Clustering

K-means

The K means algorithm is provided in the `class` package and the function is named `kmeans`

```
library(class)
#?kmeans
```

we see here that k-means 38,50,62 are not perfect solutions

```
#try 1. not set nstart, 2. try nstart=10 not a good result
#interesting in 3 group
table(x[,5])
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

```
out=kmeans(x[,-5],3)
out
```

```
## K-means clustering with 3 clusters of sizes 62, 50, 38
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.901613    2.748387     4.393548    1.433871
## 2     5.006000    3.428000     1.462000    0.246000
## 3     6.850000    3.073684     5.742105    2.071053
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [71] 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 3 3
## [106] 3 1 3 3 3 3 3 3 1 1 3 3 3 3 1 3 1 3 1 3 3 1 1 3 3 3 3 3 1 3 3 3 3 1 3
## [141] 3 3 1 3 3 3 1 3 3 1
##
## Within cluster sum of squares by cluster:
## [1] 39.82097 15.15100 23.87947
##  (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

- betweenss: is the between clusters sum of squares. In fact it is the mean of distances between cluster centers.
- totss: total some of square

```
out$betweenss
```

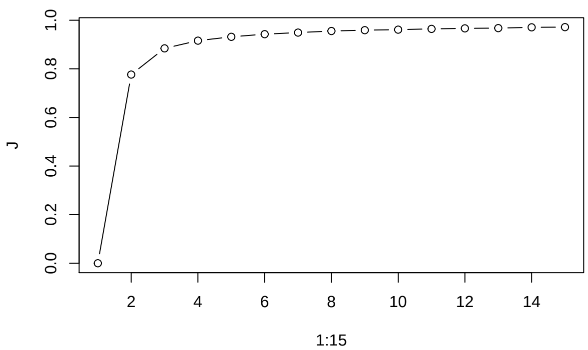
```
## [1] 602.5192
```

```
out$totss
```

```
## [1] 681.3706
```

let's try to find the most appropriate number of groups:

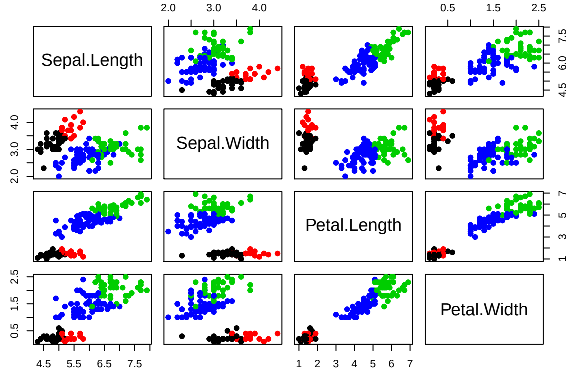
```
J=c()
for (k in 1:15){
  out=kmeans(x[,-5],k,nstart=15)
  J[k]=out$betweenss/out$totss    #B/S
}
plot(1:15,J,type='b')
```



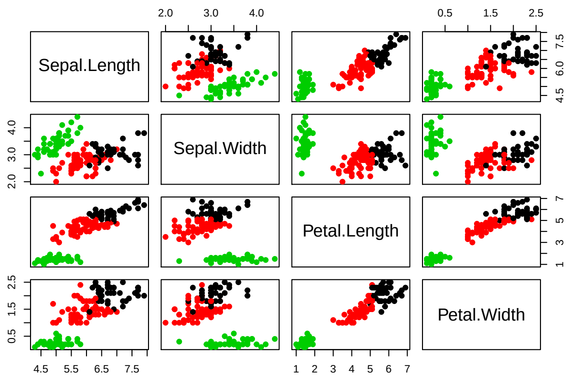

```
#we choose k from 2-15, so
```

here we shoud choose wither 3 or 4 groups

```
out= kmeans(x[,-5],4)
pairs(x[,-5],col=out$cluster,pch=19)
```



```
out1= kmeans(x[,-5],3)
pairs(x[,-5],col=out1$cluster,pch=19)
```



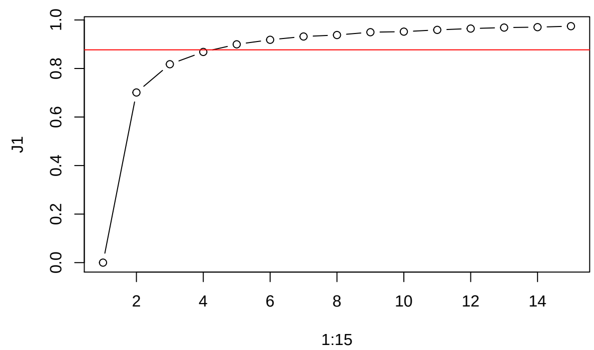
exercuse: use the k-means to cluster ‘swiss’ data

```
x1=swiss
summary(swiss)
```

```
##    Fertility      Agriculture     Examination      Education
##  Min.   :35.00   Min.   : 1.20   Min.   : 3.00   Min.   : 1.00
##  1st Qu.:64.70   1st Qu.:35.90   1st Qu.:12.00   1st Qu.: 6.00
##  Median :70.40   Median :54.10   Median :16.00   Median : 8.00
##  Mean   :70.14   Mean   :50.66   Mean   :16.49   Mean   :10.98
##  3rd Qu.:78.45   3rd Qu.:67.65   3rd Qu.:22.00   3rd Qu.:12.00
##  Max.   :92.50   Max.   :89.70   Max.   :37.00   Max.   :53.00
##     Catholic       Infant.Mortality
##  Min.   :  2.150   Min.   :10.80
##  1st Qu.:  5.195   1st Qu.:18.15
##  Median : 15.140   Median :20.00
##  Mean   : 41.144   Mean   :19.94
##  3rd Qu.: 93.125   3rd Qu.:21.70
##  Max.   :100.000   Max.   :26.60
```

```
J1=c()
for (k in 1:15){
  out=kmeans(x1,k,nstart=15)
  J1[k]=out$betweenss/out$totss    #B/S
}
plot(1:15,J1,type='b')

# we find a smallest point up to the last 10 % (we can put a threshold on the plot)
abline(h=0.9*max(J1[15]-J1[1]),col='red')
```

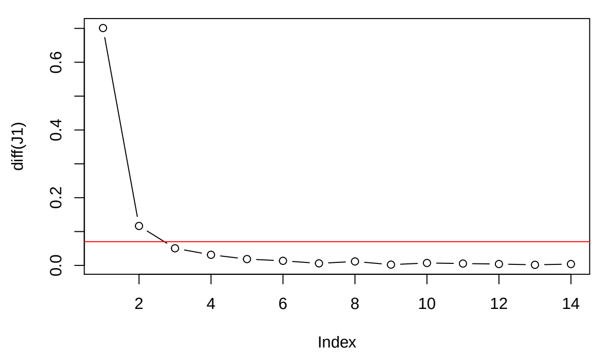



```
J1
```

```
## [1] -2.456367e-16 7.010579e-01 8.175599e-01 8.681460e-01 8.995247e-01
## [6] 9.182207e-01 9.318062e-01 9.378618e-01 9.495320e-01 9.518226e-01
## [11] 9.590178e-01 9.644932e-01 9.685741e-01 9.703573e-01 9.743515e-01
```

Better automation to find give us the optimal point with threshold=0.1

```
#better automation: point to point difference
thd=0.1
plot(diff(J1),type='b')
abline(h=thd*max(diff(J1)),col='red')
```

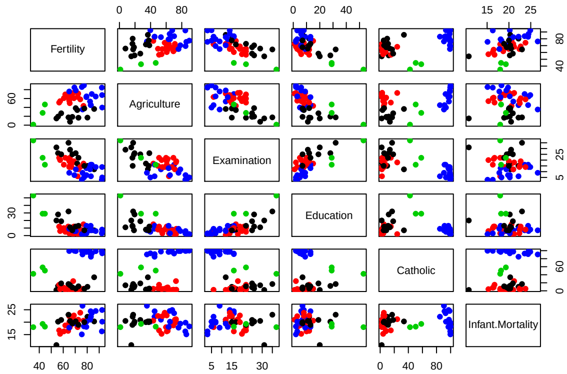



```
#plus one because the points here are the difference
#for
Kstar=max(which(diff(J1)>=thd*max(diff(J1))))+1
Kstar
```

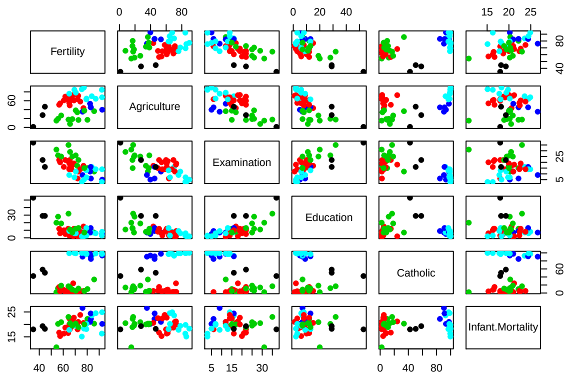
```
## [1] 3
```

I assume either 4 or 5 groups to choose

```r
out11= kmeans(x1,4,nstart=15)
pairs(x1,col=out11$cluster,pch=19)
```




```r
out12= kmeans(x1,5, nstart=15)
pairs(x1,col=out12$cluster,pch=19)
```




```r
out12
```

```
## K-means clustering with 5 clusters of sizes 3, 16, 12, 5, 11
##
## Cluster means:
##   Fertility Agriculture Examination Education Catholic Infant.Mortality
## 1  40.83333    25.16667   25.000000 37.000000 50.36667         18.50000
## 2  66.31250    60.72500   16.937500  7.687500  6.45875         19.55000
## 3  68.70000    23.80000   23.166667 14.666667 11.74333         19.71667
## 4  83.40000    43.72000    9.600000  8.200000 91.57200         22.88000
## 5  79.25455    75.42727    9.363636  5.909091 98.23091         19.81818
##
## Clustering vector:
##   Courtelary     Delemont Franches-Mnt      Moutier   Neuveville
##            3            4            4            3            2
##   Porrentruy        Broye        Glane      Gruyere       Sarine
##            4            5            5            4            4
```

```
##      Veveyse         Aigle      Aubonne     Avenches     Cossonay 
##            5             2            2            2            2 
##    Echallens      Grandson     Lausanne    La Vallee       Lavaux 
##            2             3            3            3            2 
##       Morges        Moudon        Nyone         Orbe         Oron 
##            2             2            2            2            2 
##      Payerne Paysd'enhaut        Rolle        Vevey      Yverdon 
##            2             2            2            3            2 
##      Conthey     Entremont       Herens     Martigwy      Monthey 
##            5             5            5            5            5 
##   St Maurice        Sierre         Sion       Boudry La Chauxdfnd 
##            5             5            5            3            3 
##     Le Locle     Neuchatel   Val de Ruz ValdeTravers V. De Geneve 
##            3             3            3            3            1 
##  Rive Droite  Rive Gauche 
##            1             1 
## 
## Within cluster sum of squares by cluster:
## [1] 1839.8794 2759.4449 4490.2569  552.5839 2262.4743
##  (between_SS / total_SS =  90.0 %)
## 
## Available components:
## 
## [1] "cluster"      "centers"      "totss"        "withinss"    
## [5] "tot.withinss" "betweenss"    "size"         "iter"        
## [9] "ifault"
```

```
#here we see that the cluster 2 means we think it is a big city, more balance
#there's geneve, Rive Droite, Rive Gauche
#with K-mean we don't really see which variable is most contributed
```

The hierarchical clustering

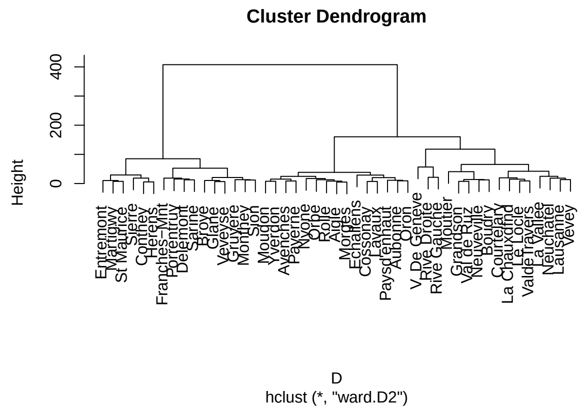
This method is implemented in R within the `class` package and the appriopriate method is named `hclust`.

> Exercise: cluster the `swiss` data with `hclust`.

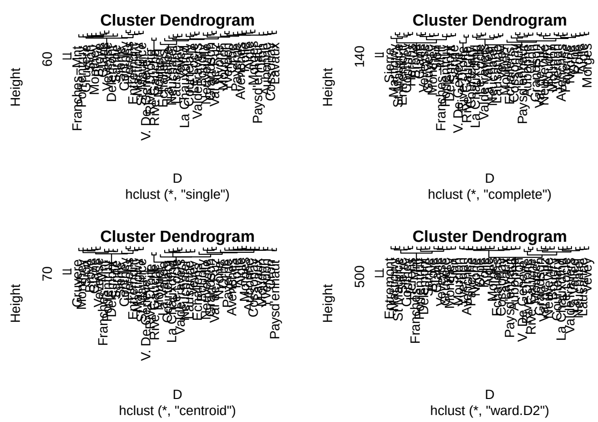
```
data(swiss)

D = dist(swiss)
out = hclust(D,method = "ward.D2")

plot(out)
```

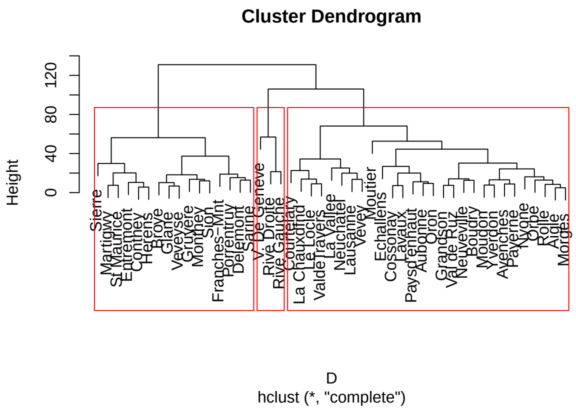



```
par(mfrow=c(2,2))
out = hclust(D,method = "single"); plot(out)
out = hclust(D,method = "complete"); plot(out)
out = hclust(D,method = "centroid"); plot(out)
out = hclust(D,method = "ward.D2"); plot(out)
```

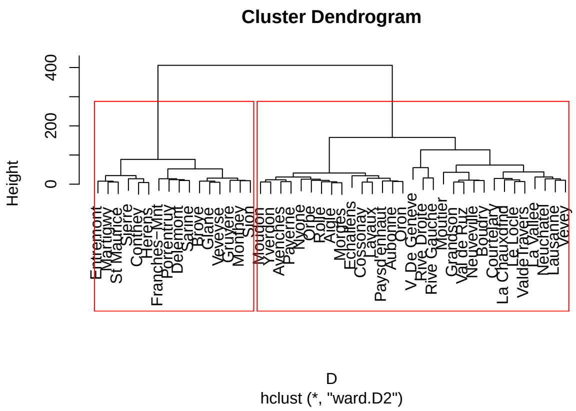


- now we choose only 2 to compare: complete and ward at this point we don't have yet the assignment to the clustering: we need cutree

```
out1 = hclust(D,method = "complete")
plot(out1)
K1 = 3
#get clustering
res1 = cutree(out1, K1)
#visualize for cluster
rect.hclust(out1,K1)
```

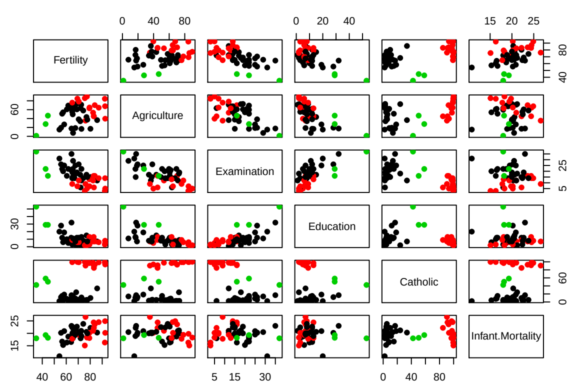



```r
out2 = hclust(D,method = "ward.D2")
plot(out2)
K2 = 2
res2 = cutree(out2, K2)
rect.hclust(out2,K2)
```

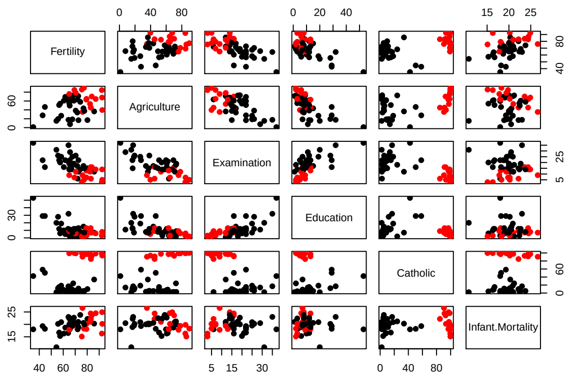


make a pair to see the variable result compaire to the clustering

```r
pairs(swiss,col = res1, pch=19)
```



```r
pairs(swiss,col = res2, pch=19)
```



The Mixture model and the EM algorithm

The mclust package (Raftery et al.) allows to cluster some datat with GMM and the EM algorithm.

```r
#install.packages('mclust')
library(mclust)
```

```
## Package 'mclust' version 5.4.3
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
data(swiss)
out = Mclust(swiss,G = 2:10)
#plot(out)
# plot 1:
#get the highest point of BIC (3 groups)
#here best model is EEE, which is exactly K-mean


# plot 2:
# we see that the result is exactly the same as HC (complete)

#plot 3:
# the larger is the point the larger is the uncertainty
```

```r
out$modelName
```

```
## [1] "EEE"
```

```r
out$parameters$mean
```

```
##                        [,1]     [,2]     [,3]
## Fertility        67.335714 80.55000 40.83333
## Agriculture      44.900000 65.51875 25.16667
```

```
## Examination      19.607143  9.43750 25.00000
## Education        10.678571  6.62500 37.00000
## Catholic          8.723571 96.15000 50.36667
## Infant.Mortality 19.621429 20.77500 18.50000
```

```
out$parameters$pro
```

```
## [1] 0.59574468 0.34042553 0.06382979
```

```
out$parameters$variance
```

```
## $modelName
## [1] "EEE"
##
## $d
## [1] 6
##
## $G
## [1] 3
##
## $sigma
## , , 1
##
##                   Fertility Agriculture Examination    Education
## Fertility         56.324063  -11.930674 -16.8778115 -14.42933131
## Agriculture      -11.930674  368.415980 -61.5283245 -73.35079786
## Examination      -16.877812  -61.528324  34.9492781  28.40615501
## Education        -14.429331  -73.350798  28.4061550  40.76291793
## Catholic           4.581123   -2.658348   0.7508359   0.02387538
## Infant.Mortality   8.648693  -11.600266   0.7853343   0.84984802
##                     Catholic Infant.Mortality
## Fertility         4.58112258        8.6486930
## Agriculture      -2.65834751      -11.6002660
## Examination       0.75083587        0.7853343
## Education         0.02387538        0.8498480
## Catholic         40.66932150       -0.0764924
## Infant.Mortality -0.07649240        7.8731307
##
## , , 2
##
##                   Fertility Agriculture Examination    Education
## Fertility         56.324063  -11.930674 -16.8778115 -14.42933131
## Agriculture      -11.930674  368.415980 -61.5283245 -73.35079786
## Examination      -16.877812  -61.528324  34.9492781  28.40615501
## Education        -14.429331  -73.350798  28.4061550  40.76291793
## Catholic           4.581123   -2.658348   0.7508359   0.02387538
## Infant.Mortality   8.648693  -11.600266   0.7853343   0.84984802
##                     Catholic Infant.Mortality
## Fertility         4.58112258        8.6486930
## Agriculture      -2.65834751      -11.6002660
## Examination       0.75083587        0.7853343
## Education         0.02387538        0.8498480
## Catholic         40.66932150       -0.0764924
```

```
## Infant.Mortality -0.07649240        7.8731307
##
## , , 3
##
##                   Fertility Agriculture Examination    Education
## Fertility         56.324063  -11.930674 -16.8778115 -14.42933131
## Agriculture      -11.930674  368.415980 -61.5283245 -73.35079786
## Examination      -16.877812  -61.528324  34.9492781  28.40615501
## Education        -14.429331  -73.350798  28.4061550  40.76291793
## Catholic           4.581123   -2.658348   0.7508359   0.02387538
## Infant.Mortality   8.648693  -11.600266   0.7853343   0.84984802
##                     Catholic Infant.Mortality
## Fertility         4.58112258        8.6486930
## Agriculture      -2.65834751      -11.6002660
## Examination       0.75083587        0.7853343
## Education         0.02387538        0.8498480
## Catholic         40.66932150       -0.0764924
## Infant.Mortality -0.07649240        7.8731307
##
##
## $Sigma
##                   Fertility Agriculture Examination    Education
## Fertility         56.324063  -11.930674 -16.8778115 -14.42933131
## Agriculture      -11.930674  368.415980 -61.5283245 -73.35079786
## Examination      -16.877812  -61.528324  34.9492781  28.40615501
## Education        -14.429331  -73.350798  28.4061550  40.76291793
## Catholic           4.581123   -2.658348   0.7508359   0.02387538
## Infant.Mortality   8.648693  -11.600266   0.7853343   0.84984802
##                     Catholic Infant.Mortality
## Fertility         4.58112258        8.6486930
## Agriculture      -2.65834751      -11.6002660
## Examination       0.75083587        0.7853343
## Education         0.02387538        0.8498480
## Catholic         40.66932150       -0.0764924
## Infant.Mortality -0.07649240        7.8731307
##
## $cholSigma
##                  Fertility Agriculture Examination Education    Catholic
## Fertility         7.504936    -1.58971   -2.248895 -1.922646  0.61041462
## Agriculture       0.000000    19.12822   -3.403527 -3.994478 -0.08824476
## Examination       0.000000     0.00000   -4.278756 -2.450949 -0.42611701
## Education         0.000000     0.00000    0.000000 -3.886303  0.05130754
## Catholic          0.000000     0.00000    0.000000  0.000000 -6.33282877
## Infant.Mortality  0.000000     0.00000    0.000000  0.000000  0.00000000
##                  Infant.Mortality
## Fertility              1.15240065
## Agriculture           -0.51067390
## Examination           -0.38302481
## Education             -0.02234927
## Catholic               0.15586488
## Infant.Mortality      -2.47241061
```

The `Rmixmod` package also allows to use the GGM + EM:

```r
#install.packages('Rmixmod')
library(Rmixmod)
```

```
## Loading required package: Rcpp
```

```
## Rmixmod v. 2.1.2.2 / URI: www.mixmod.org
```

```r
out = mixmodCluster(swiss,2:10)
# 2:10 = means that it would choose the best group between it
#default is 1 to 9. if it's 1 it means that there's no need to do clustering
#plot(out) # type in the console
```

(short insert cut ctrl+alt+I)