D1 K Mean

reminder about the descriptive statisic

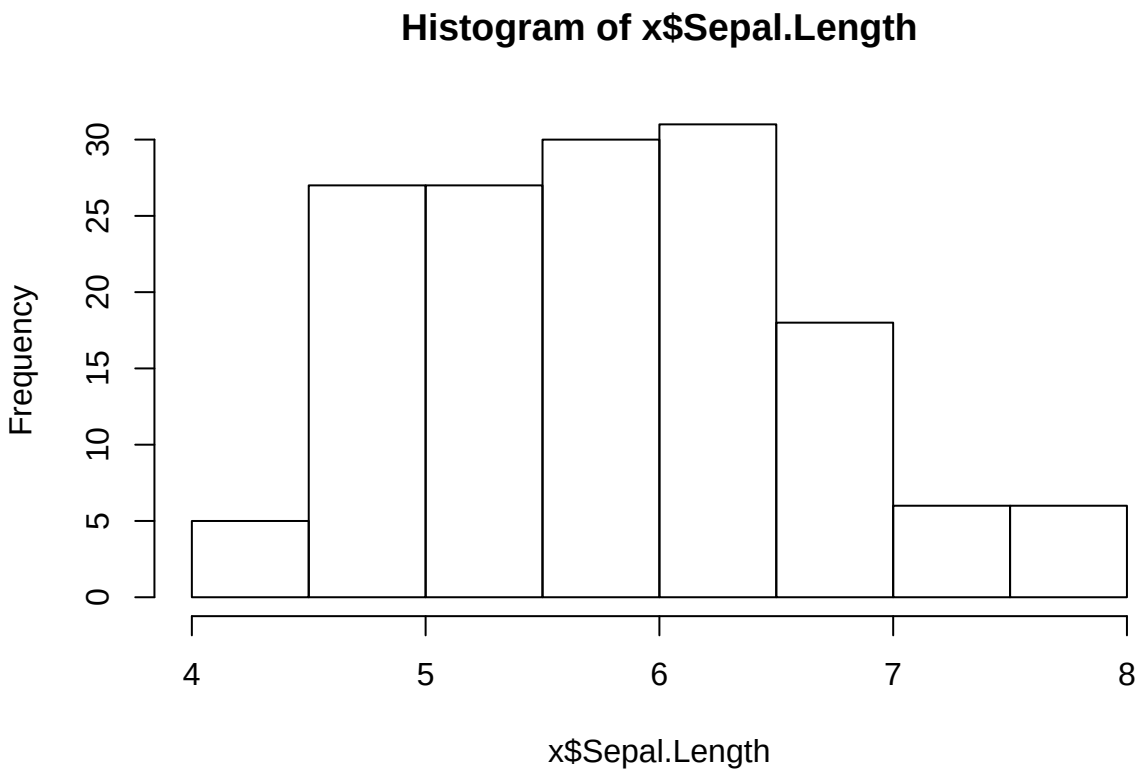
```
x=iris
summary(x)
```

```
##   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 (bins number)

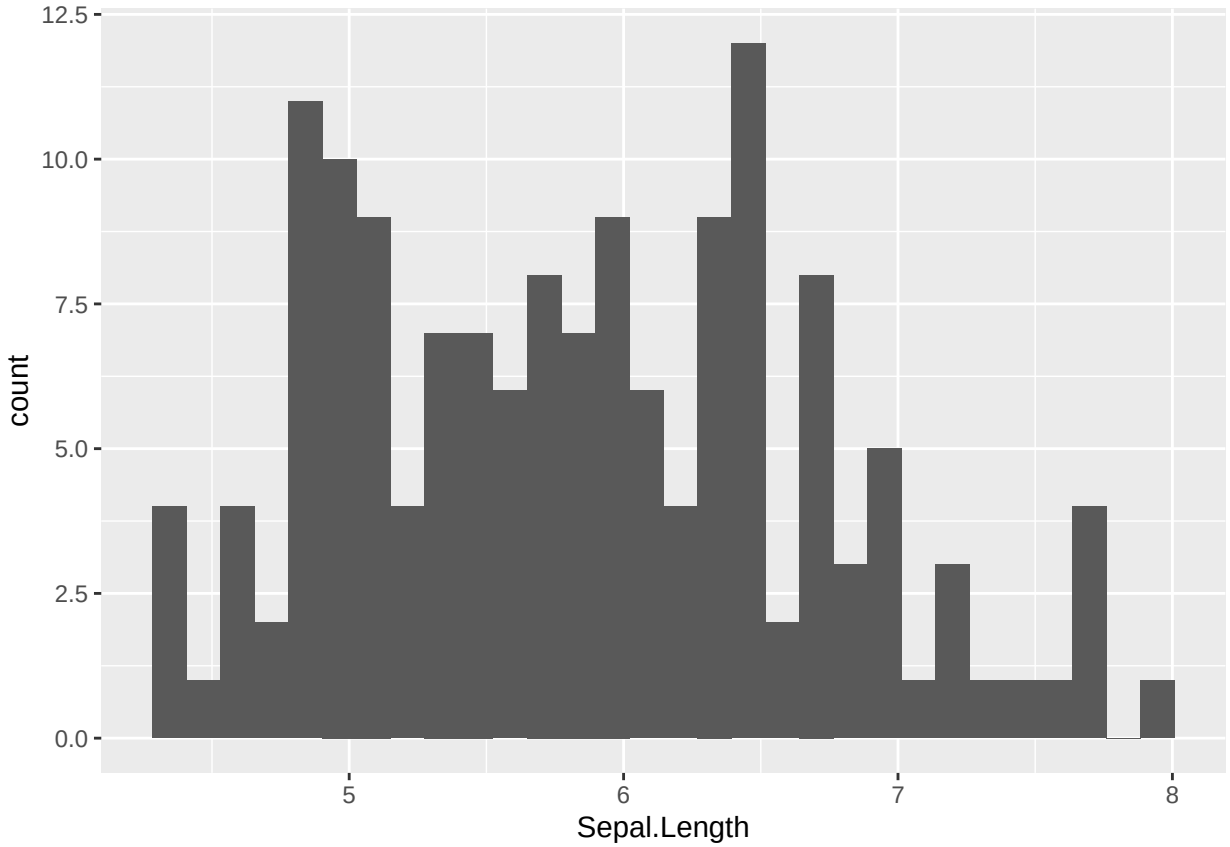
```
hist(x$Sepal.Length)
```



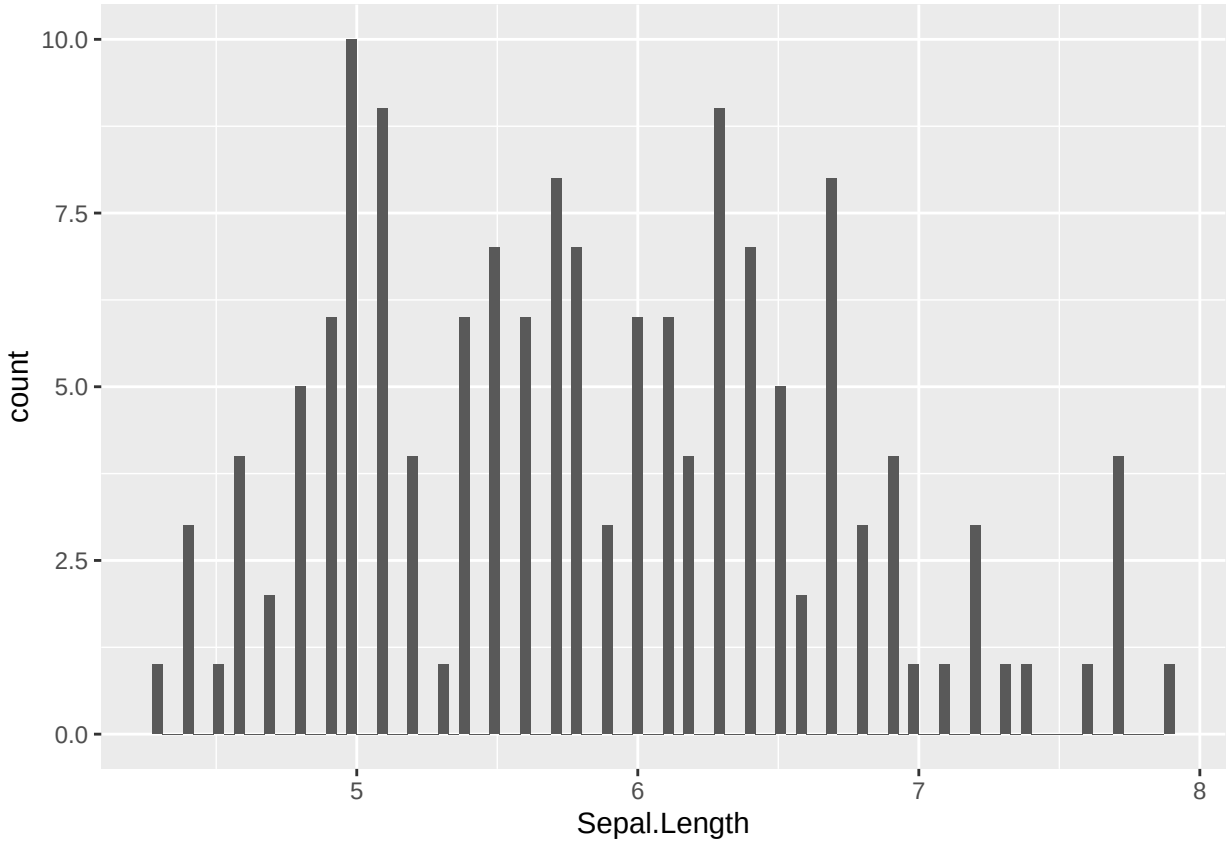
- best choice by ggplot (bins number)

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

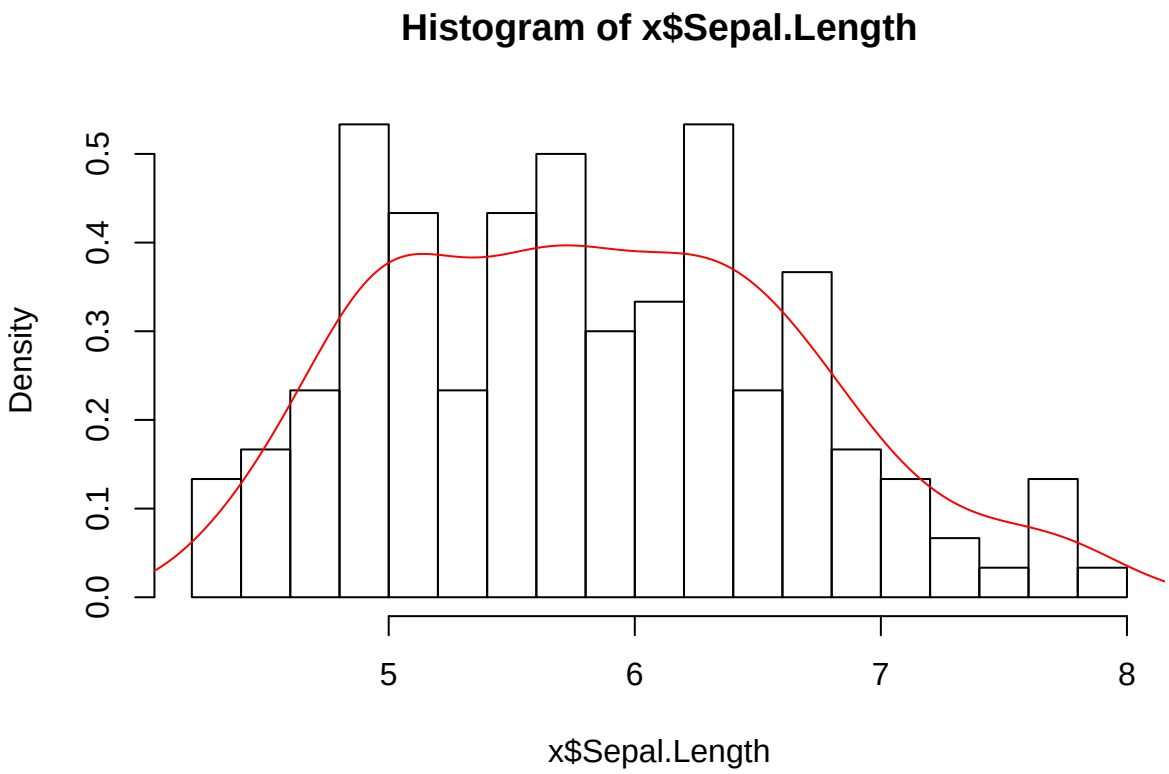
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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

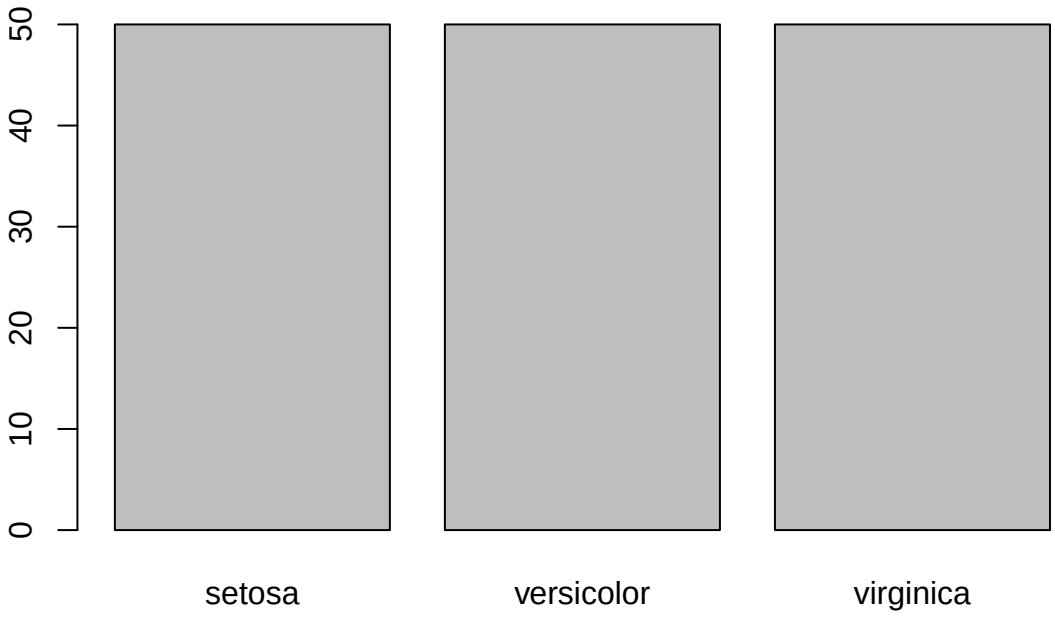



```
hist(x$Sepal.Length,breaks=20,freq=FALSE)
lines(density(x$Sepal.Length),col='red')
```



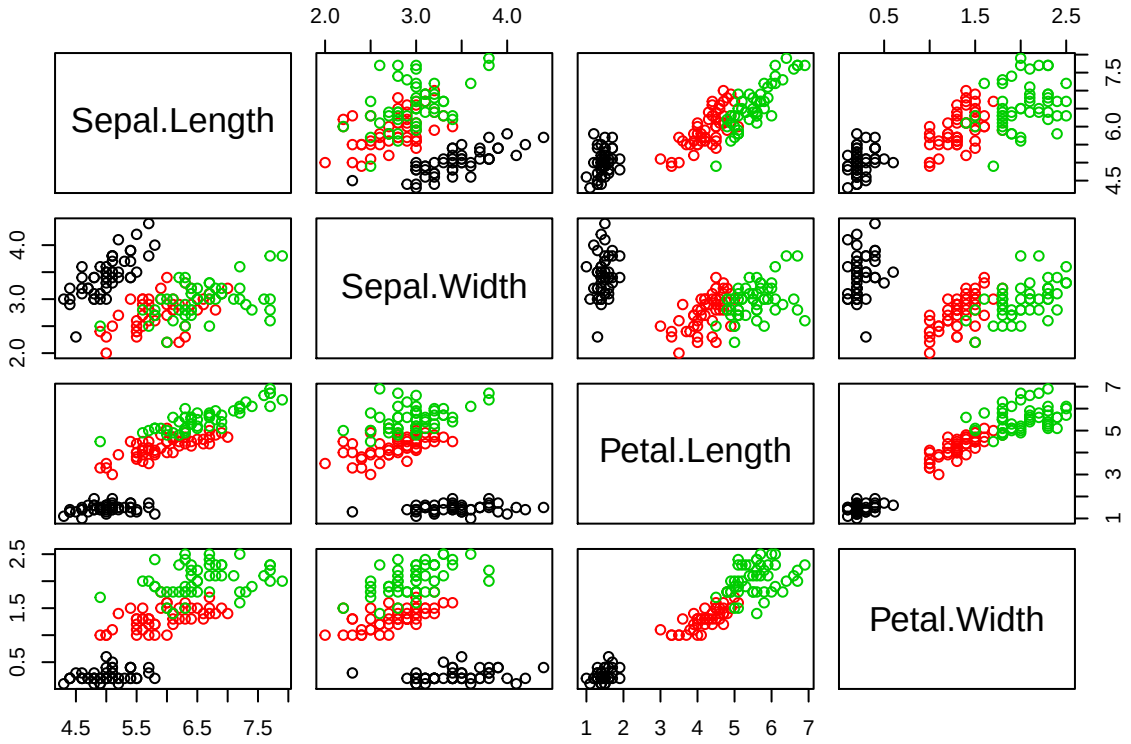
- boxplot check petal.length, here we have a distribution, very low to increase and very fast to decrease

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



* try to pair to see the relationship

```
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
```

```
## starting httpd help server ... done
```

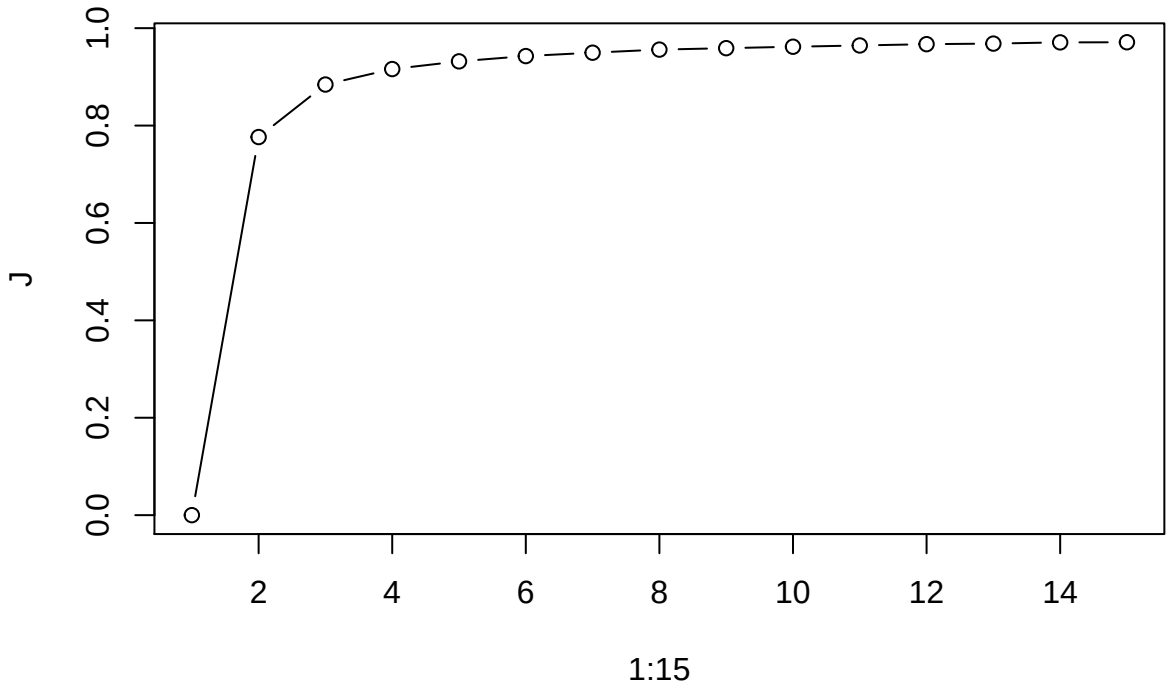
```
#try 1. not set nstart, 2. try nstart=10 not a good result
out=kmeans(x[,-5],3,nstart=10)
out
```

```
## K-means clustering with 3 clusters of sizes 62, 38, 50
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.901613    2.748387     4.393548    1.433871
## 2     6.850000    3.073684     5.742105    2.071053
## 3     5.006000    3.428000     1.462000    0.246000
##
```

```
## Clustering vector:
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [36] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [71] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2
## [106] 2 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2 1 2
## [141] 2 2 1 2 2 2 1 2 2 1
##
## Within cluster sum of squares by cluster:
## [1] 39.82097 23.87947 15.15100
##  (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

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

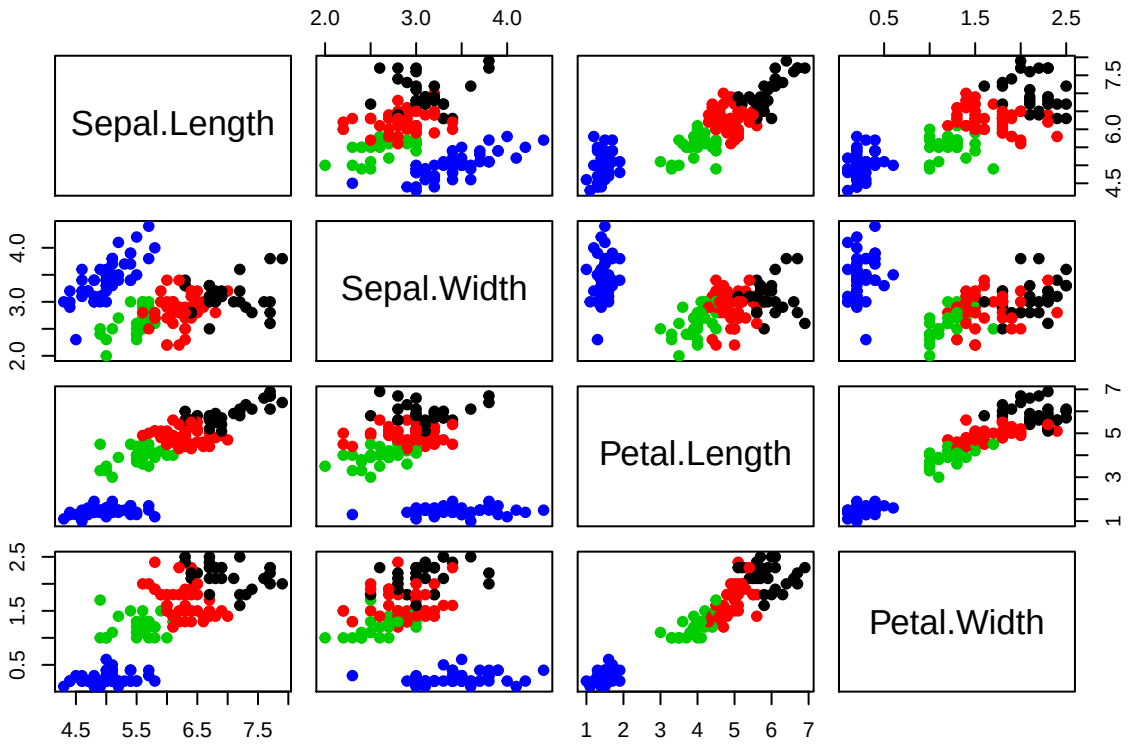
```
J=c()
for (k in 1:15){
  out=kmeans(x[,-5],k,nstart=15)
  J[k]=out$betweenss/out$totss   #B/S
}
plot(1:15,J,type='b')
```



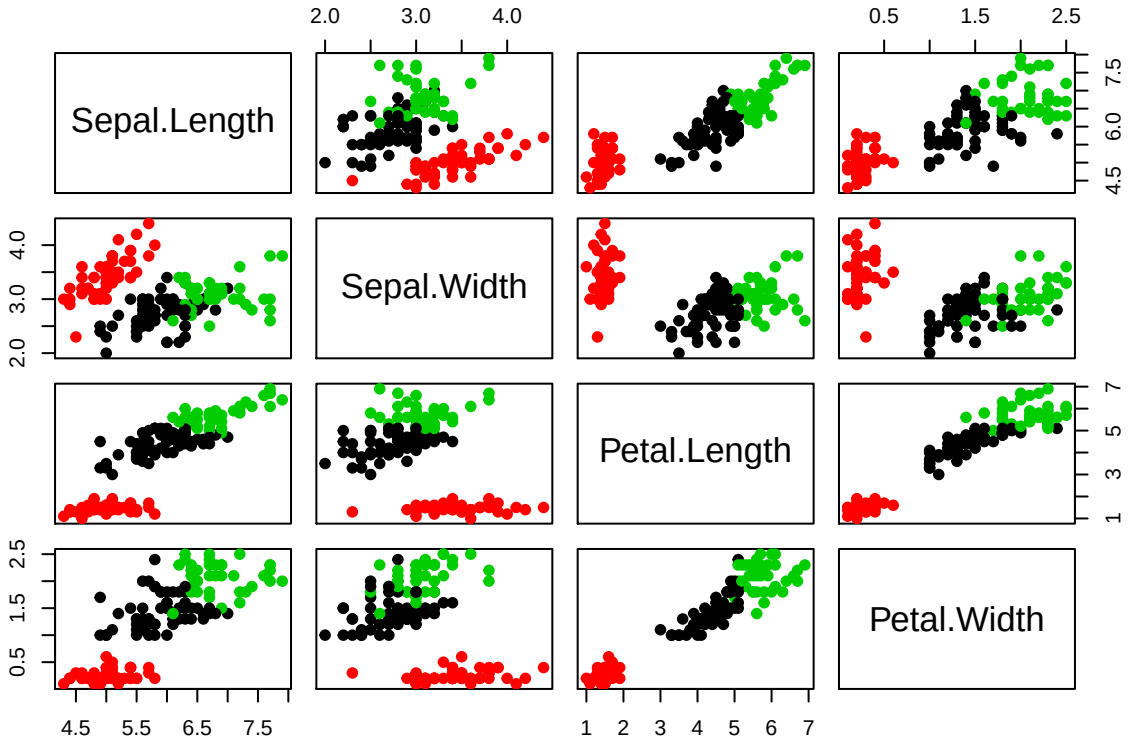
```
#we choose k from 2-15, so
```

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

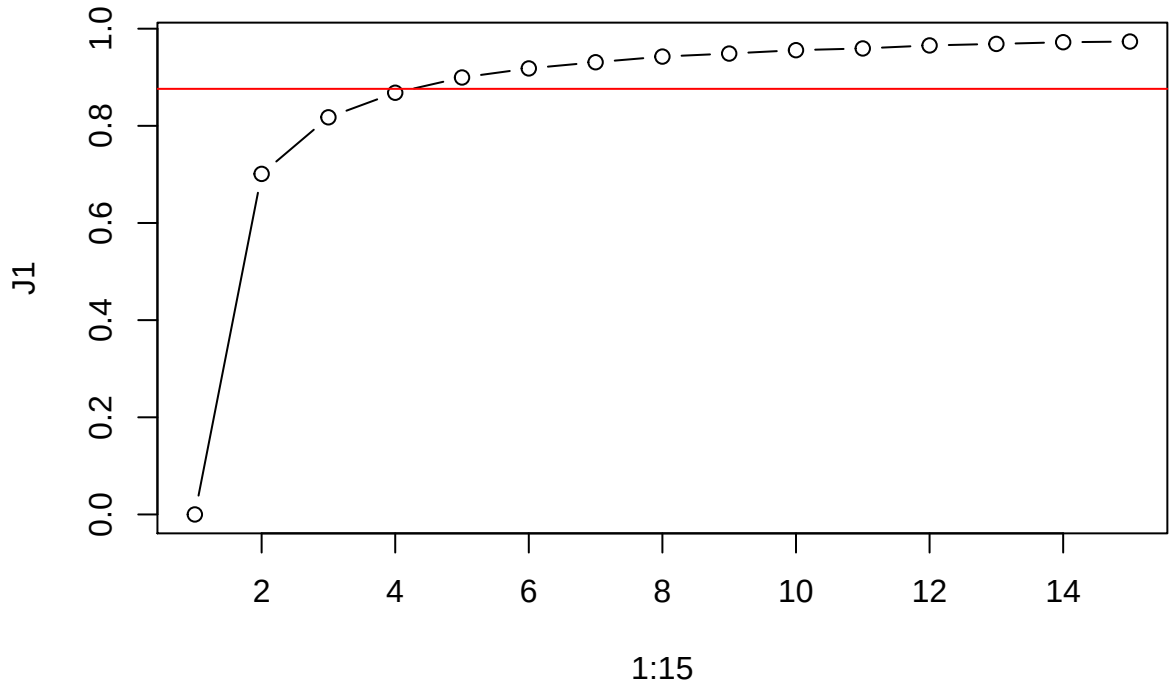
```
x1=swiss
summary(swiss)
```

```
##    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
}
```

```r
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')
```

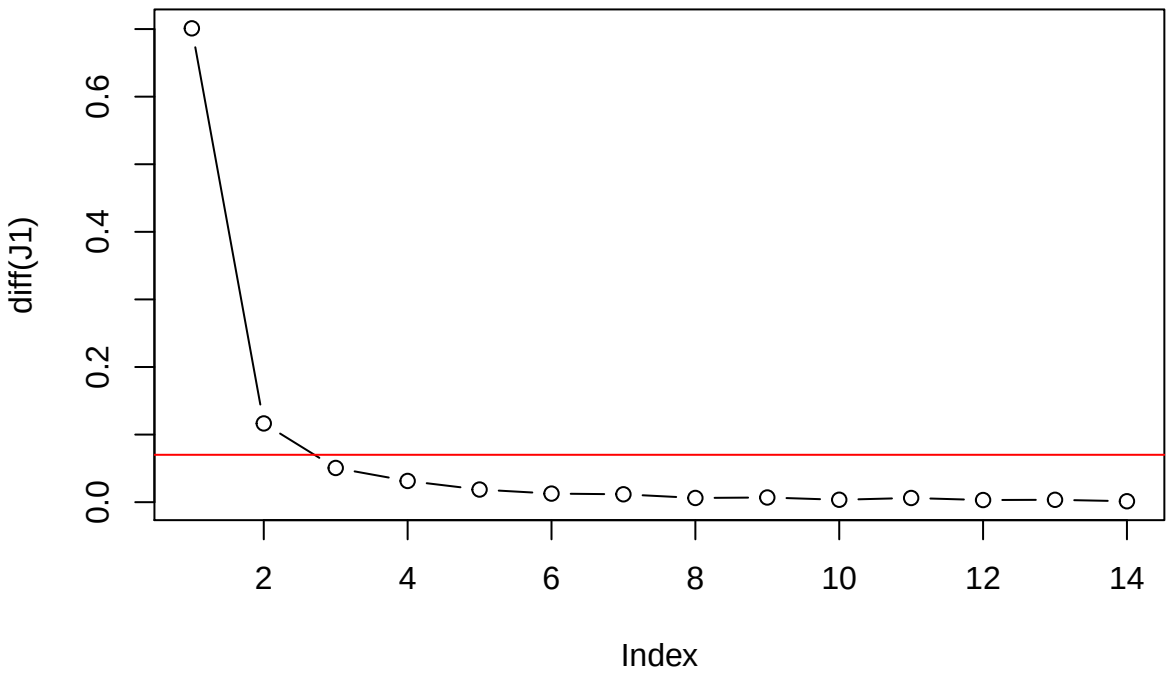



```r
J1
```

```
##  [1] -2.456367e-16  7.010579e-01  8.175599e-01  8.681460e-01  8.995247e-01
##  [6]  9.182207e-01  9.309163e-01  9.425865e-01  9.487977e-01  9.557432e-01
## [11]  9.592991e-01  9.654925e-01  9.686482e-01  9.721368e-01  9.735147e-01
```

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

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

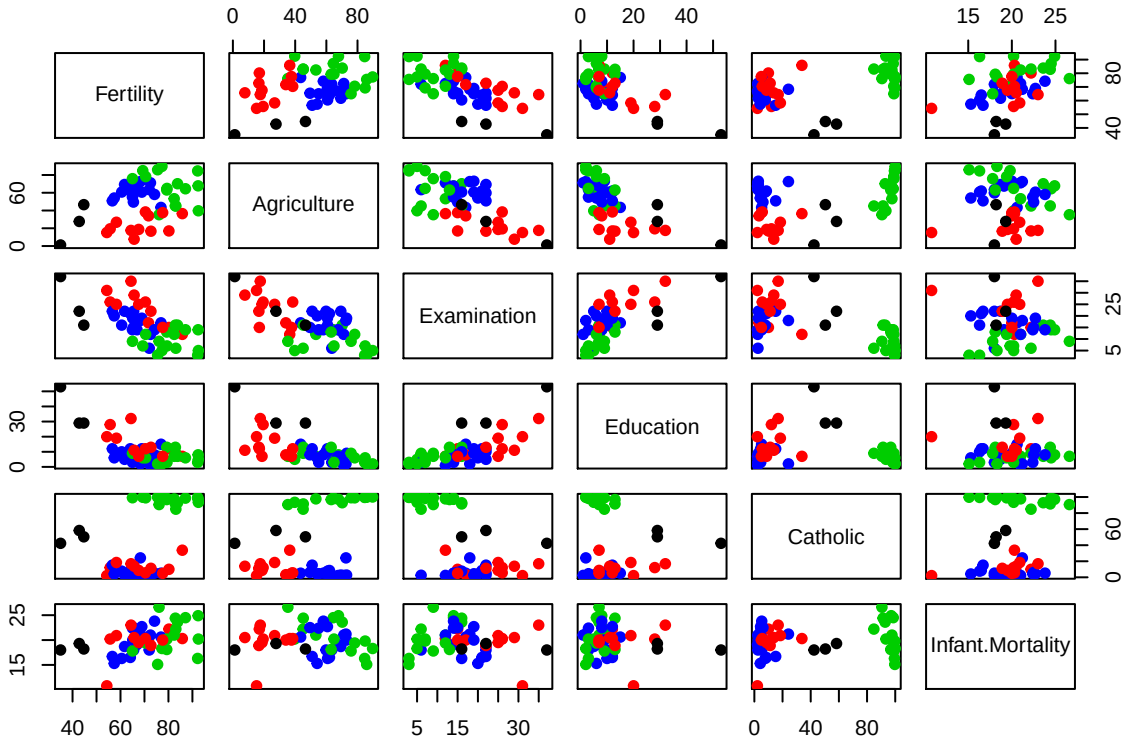



```
#plus one because the points here are the difference
#for
Kstar=max(which(diff(J1)>=thd*max(diff(J1))))+1
Kstar
```

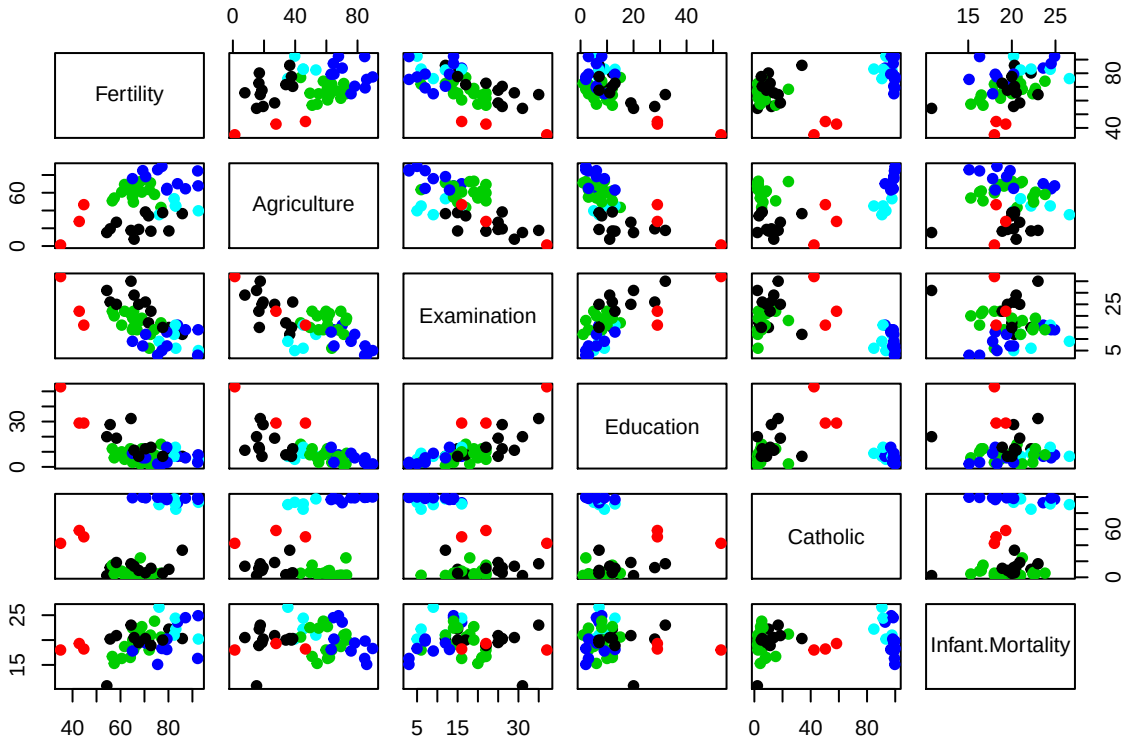
```
## [1] 3
```

I assume either 4 or 5 groups to choose

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




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




```
out12
```

```
## K-means clustering with 5 clusters of sizes 12, 3, 16, 11, 5
##
## Cluster means:
##   Fertility Agriculture Examination Education Catholic Infant.Mortality
## 1  68.70000    23.80000   23.166667 14.666667 11.74333         19.71667
## 2  40.83333    25.16667   25.000000 37.000000 50.36667         18.50000
## 3  66.31250    60.72500   16.937500  7.687500  6.45875         19.55000
## 4  79.25455    75.42727    9.363636  5.909091 98.23091         19.81818
## 5  83.40000    43.72000    9.600000  8.200000 91.57200         22.88000
##
## Clustering vector:
##   Courtelary     Delemont Franches-Mnt      Moutier   Neuveville
##            1            5            5            1            3
##   Porrentruy        Broye        Glane      Gruyere       Sarine
##            5            4            4            5            5
##      Veveyse        Aigle      Aubonne     Avenches     Cossonay
##            4            3            3            3            3
##    Echallens     Grandson     Lausanne    La Vallee       Lavaux
##            3            1            1            1            3
##       Morges       Moudon        Nyone         Orbe         Oron
##            3            3            3            3            3
##      Payerne Paysd'enhaut        Rolle        Vevey      Yverdon
##            3            3            3            1            3
##      Conthey    Entremont       Herens     Martigwy      Monthey
```

```
##            4            4            4            4            4
##   St Maurice       Sierre         Sion       Boudry La Chauxdfnd
##            4            4            4            1            1
##     Le Locle    Neuchatel   Val de Ruz ValdeTravers V. De Geneve
##            1            1            1            1            2
##  Rive Droite  Rive Gauche
##            2            2
##
## Within cluster sum of squares by cluster:
## [1] 4490.2569 1839.8794 2759.4449 2262.4743  552.5839
##  (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
```

(short insert cut ctrl+alt+I)