UnsupervisedLearning

July 15, 2020

1 Clustering with K-means

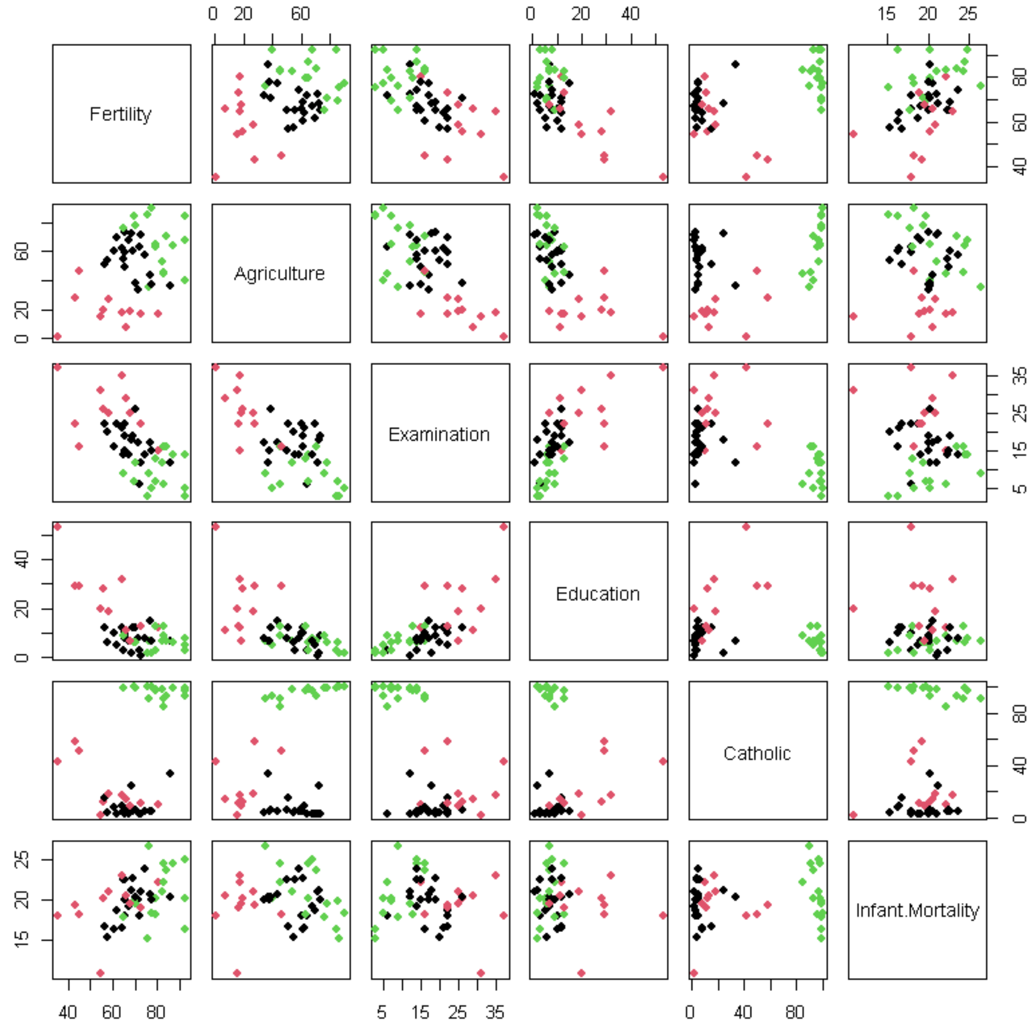
1.1 With swiss data

```
K-means clustering with 3 clusters of sizes 16, 20, 11

Cluster means:
  Fertility Agriculture Examination Education Catholic Infant.Mortality
1  80.55000    65.51875     9.43750     6.625 96.15000         20.77500
2  68.32500    55.90500    17.05000     7.850  7.55000         19.67000
3  58.30909    19.50909    25.72727    23.000 22.21455         19.22727

Clustering vector:
  Courtelary     Delemont Franches-Mnt      Moutier   Neuveville   Porrentruy 
           3            1            1            2            2            1 
       Broye        Glane      Gruyere       Sarine      Veveyse        Aigle 
           1            1            1            1            1            2 
     Aubonne     Avenches     Cossonay    Echallens     Grandson     Lausanne 
           2            2            2            2            2            3 
   La Vallee       Lavaux       Morges       Moudon        Nyone         Orbe 
           3            2            2            2            2            2 
        Oron      Payerne Paysd'enhaut        Rolle        Vevey      Yverdon 
           2            2            2            2            3            2 
     Conthey    Entremont       Herens     Martigwy      Monthey   St Maurice 
           1            1            1            1            1            1 
      Sierre         Sion       Boudry La Chauxdfnd     Le Locle    Neuchatel 
           1            1            2            3            3            3 
  Val de Ruz ValdeTravers V. De Geneve  Rive Droite  Rive Gauche 
           2            3            3            3            3 

Within cluster sum of squares by cluster:
[1] 6532.906 5966.297 9116.894
 (between_SS / total_SS =  81.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

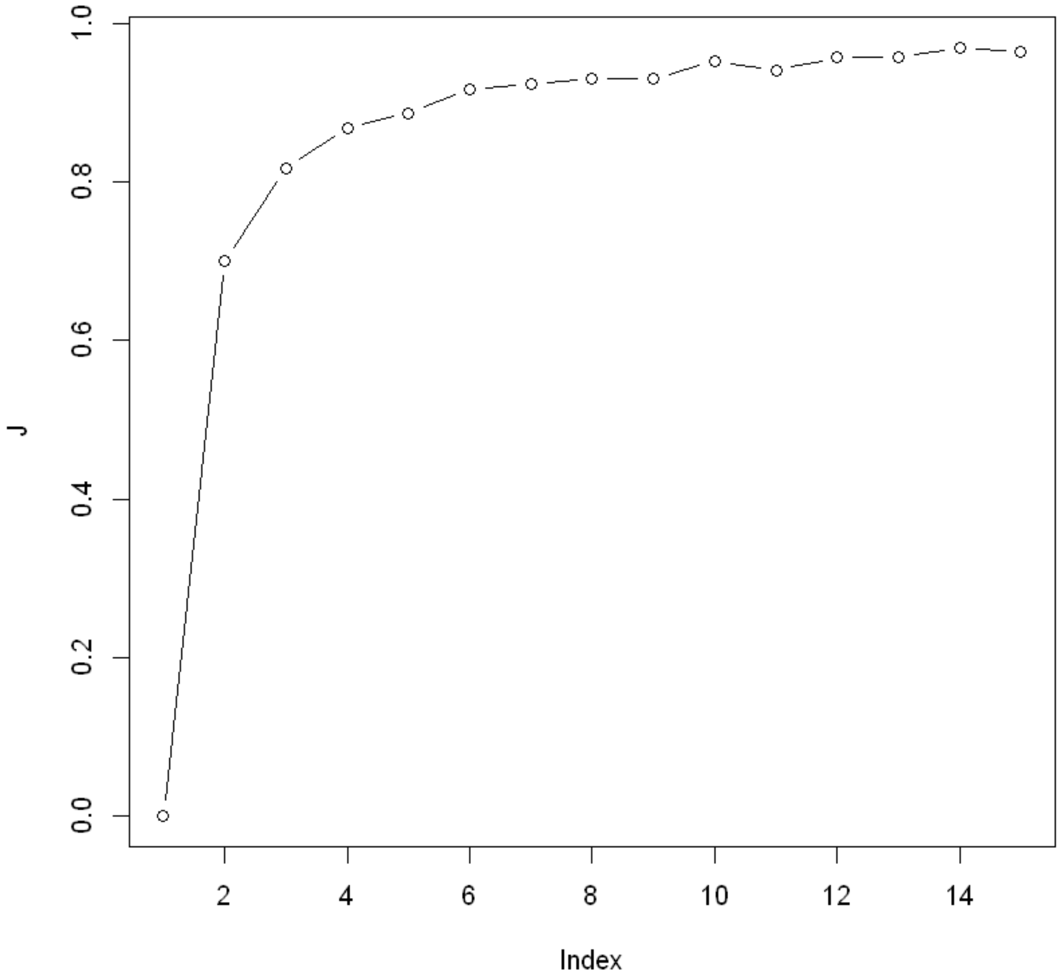
- first size of each group
- then the means of each cluster for each variable = "average guy" for each cluster.
- then assignement to each individuals to each group
- finally the g(K) value.

21616.0973659091

0.817559896131908

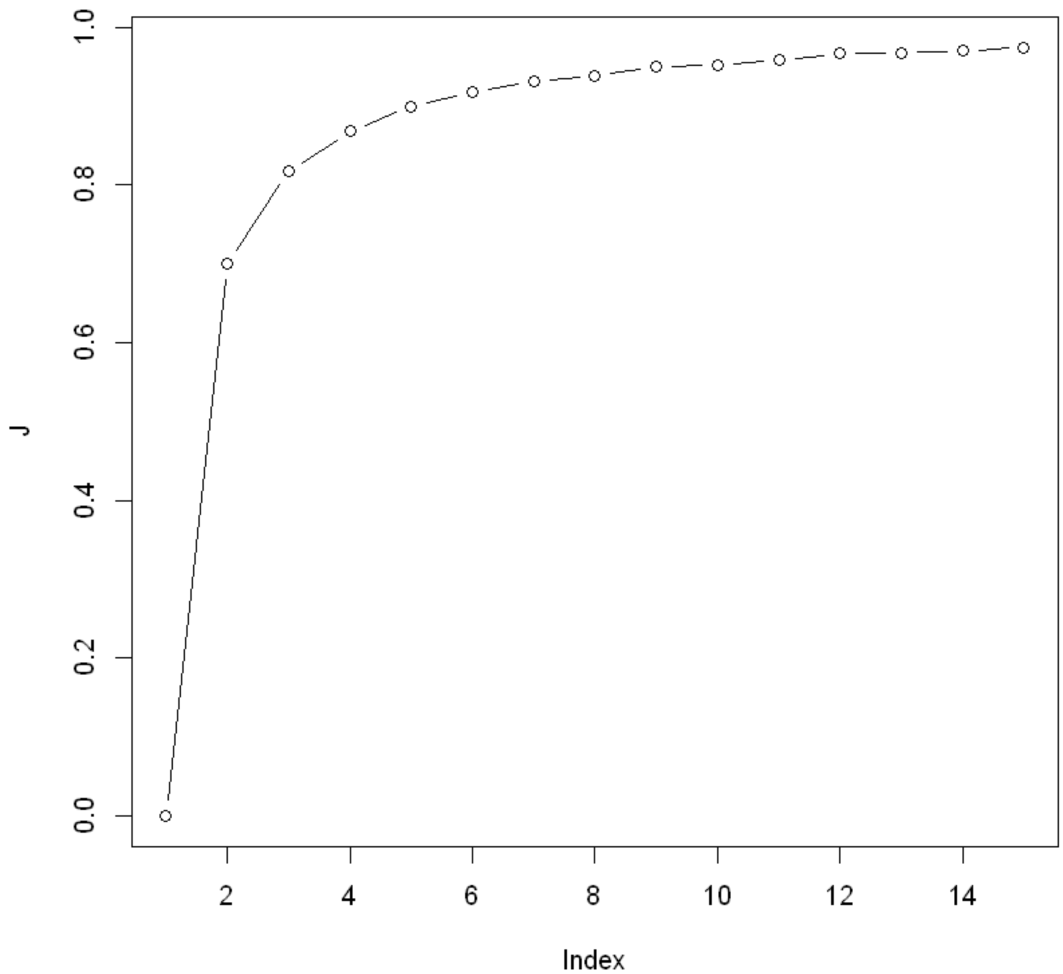


- one cluster for "big" cities
- one cluster for catholic cities
- one cluster for protestant cities



The curve sometime decrease, because the alogrythms change its init points at each time.

The solution is to run several time with the same k : nstart = 10



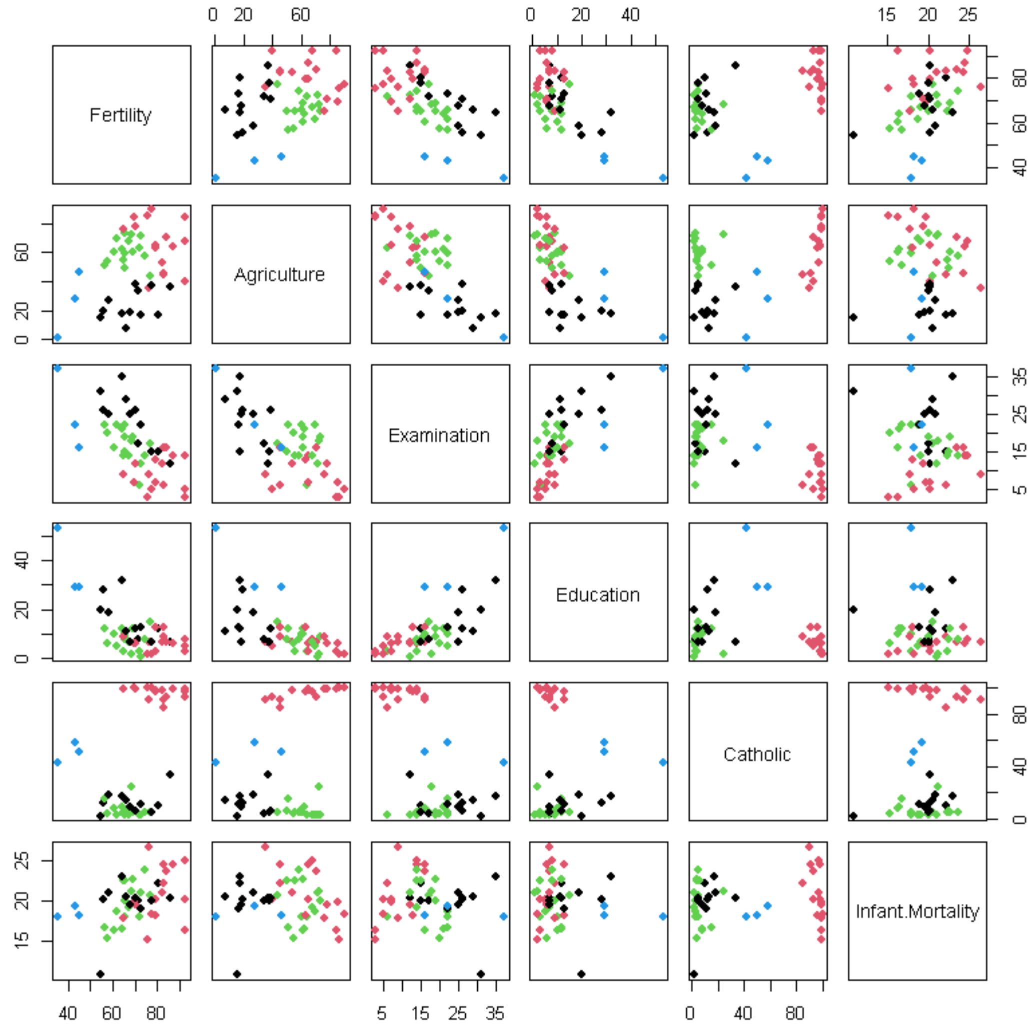
The choice could be K = 4

When we run ten times (nstart = 10) the result is the clustering with the best j(K)

```
K-means clustering with 4 clusters of sizes 12, 16, 16, 3

Cluster means:
  Fertility Agriculture Examination Education Catholic Infant.Mortality
1  68.70000    23.80000    23.16667  14.66667 11.74333         19.71667
2  80.55000    65.51875     9.43750   6.62500 96.15000         20.77500
3  66.31250    60.72500    16.93750   7.68750  6.45875         19.55000
4  40.83333    25.16667    25.00000  37.00000 50.36667         18.50000

Clustering vector:
```

```
  Courtelary     Delemont Franches-Mnt      Moutier   Neuveville   Porrentruy 
           1            2            2            1            3            2 
       Broye        Glane      Gruyere       Sarine      Veveyse        Aigle 
           2            2            2            2            2            3 
     Aubonne     Avenches     Cossonay    Echallens     Grandson     Lausanne 
           3            3            3            3            1            1 
   La Vallee       Lavaux       Morges       Moudon        Nyone         Orbe 
           1            3            3            3            3            3 
        Oron      Payerne Paysd'enhaut        Rolle        Vevey      Yverdon 
           3            3            3            3            1            3 
     Conthey    Entremont       Herens     Martigwy      Monthey   St Maurice 
           2            2            2            2            2            2 
      Sierre         Sion       Boudry La Chauxdfnd     Le Locle    Neuchatel 
           2            2            1            1            1            1 
  Val de Ruz ValdeTravers V. De Geneve  Rive Droite  Rive Gauche 
           1            1            4            4            4 

Within cluster sum of squares by cluster:
[1] 4490.257 6532.906 2759.445 1839.879
 (between_SS / total_SS =  86.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"      
```

Now the clustering means : * catholic * protestant * big cities : 3 * protestant , no agriculture and protestant

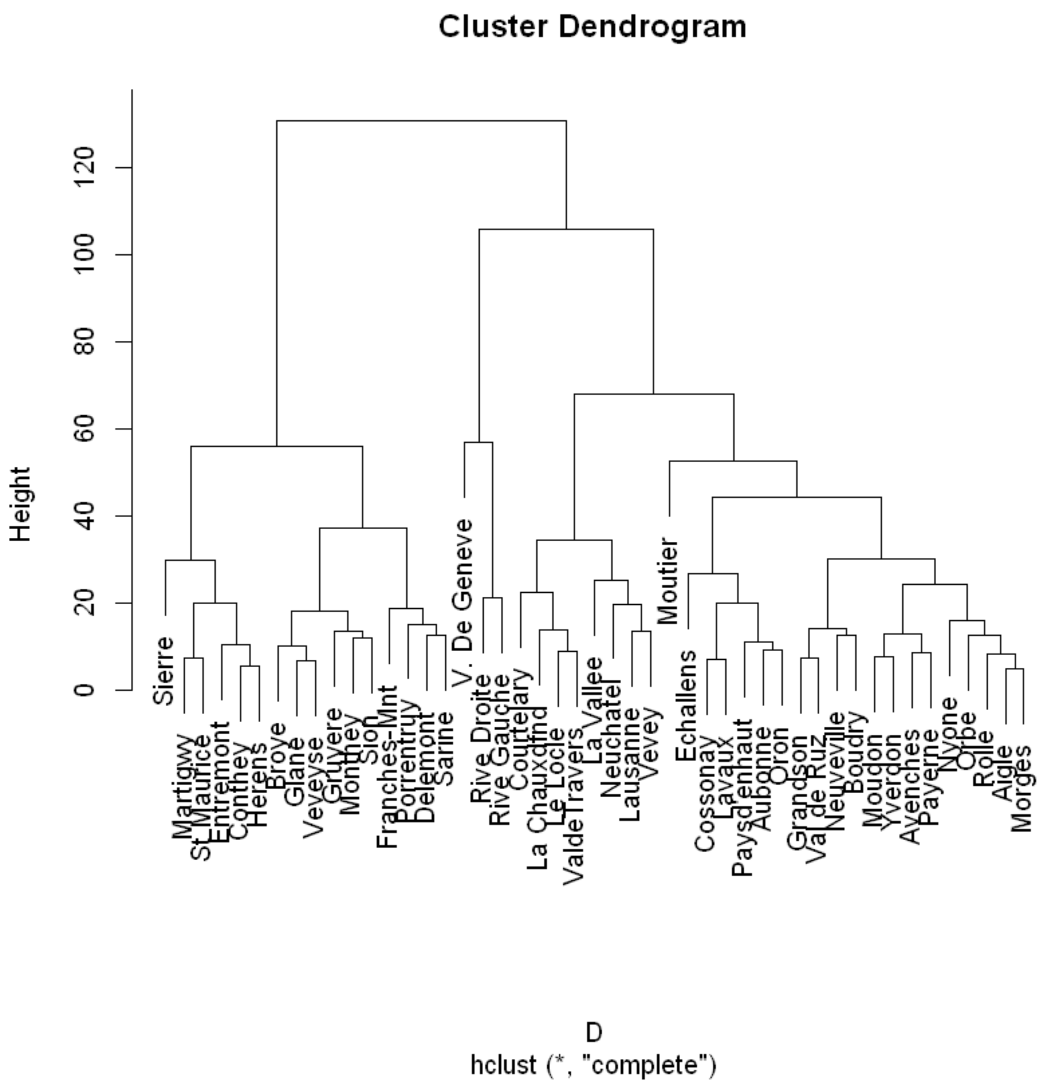


2 Hierarchical clustering

The input data is not the X variable but a matrices with the distances between the observations

```
Call:
hclust(d = D, method = "complete")

Cluster method   : complete
Distance         : euclidean
Number of objects: 47
```

To see the result we have to plot the Dendogram



Here when looking at the Dendogram we may choose to cut the tree at k=3 because it's there that the step is the biggest.

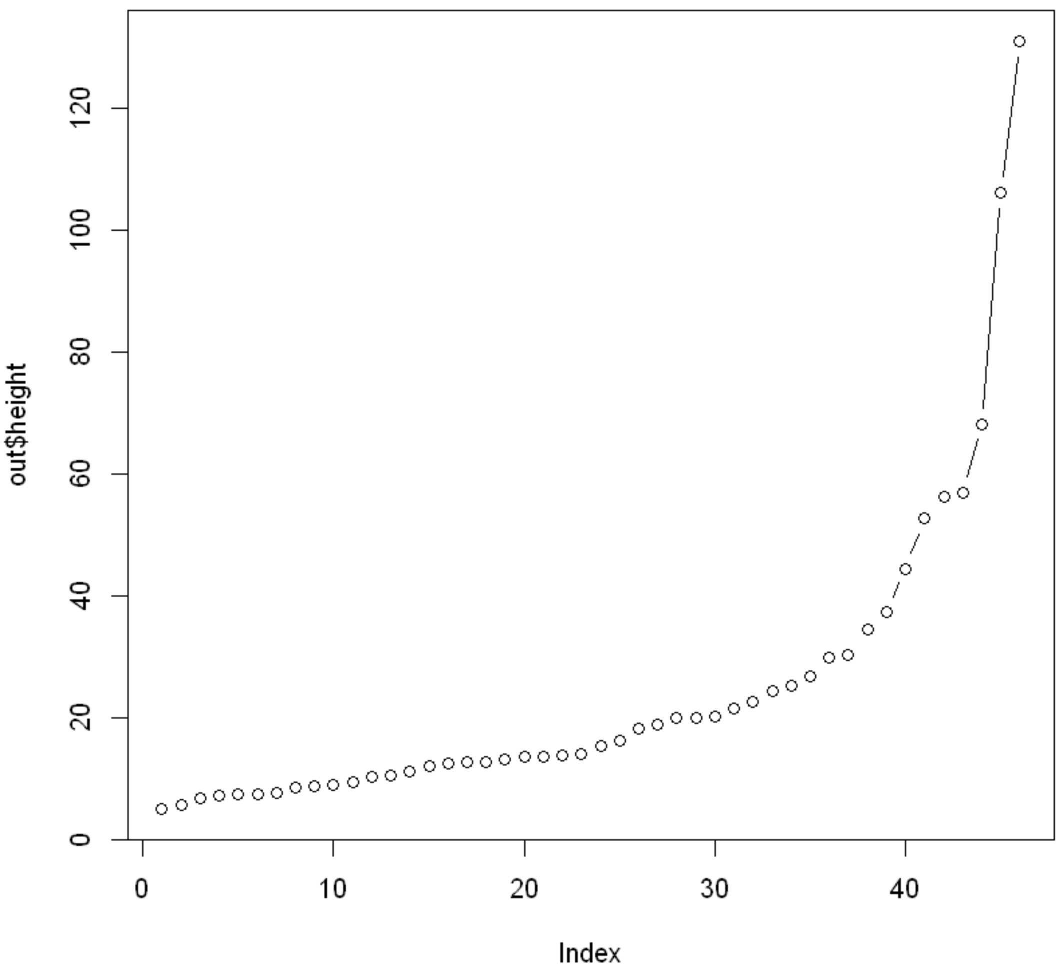
In order to obtain the clustering final partition, we have to cut the tree at the level k=3

```
  Courtelary     Delemont Franches-Mnt      Moutier  Neuveville   Porrentruy
           1            2            2            1            1            2
       Broye        Glane      Gruyere       Sarine      Veveyse        Aigle
           2            2            2            2            2            1
     Aubonne     Avenches     Cossonay    Echallens     Grandson     Lausanne
           1            1            1            1            1            1
   La Vallee       Lavaux       Morges       Moudon        Nyone         Orbe
           1            1            1            1            1            1
```

```
        Oron      Payerne Paysd'enhaut        Rolle        Vevey      Yverdon
           1            1            1            1            1            1
     Conthey    Entremont       Herens     Martigwy      Monthey   St Maurice
           2            2            2            2            2            2
      Sierre         Sion       Boudry La Chauxdfnd     Le Locle    Neuchatel
           2            2            1            1            1            1
  Val de Ruz ValdeTravers V. De Geneve  Rive Droite  Rive Gauche
           1            1            3            3            3
```

We can also use the numeric values in the out to draw a similar curse as for k-means

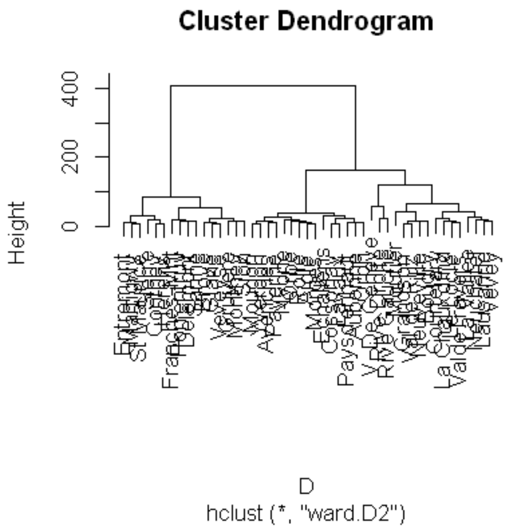
1. 'merge' 2. 'height' 3. 'order' 4. 'labels' 5. 'method' 6. 'call' 7. 'dist.method'

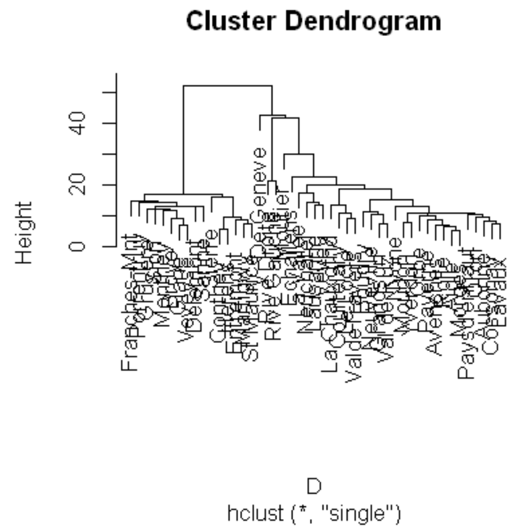


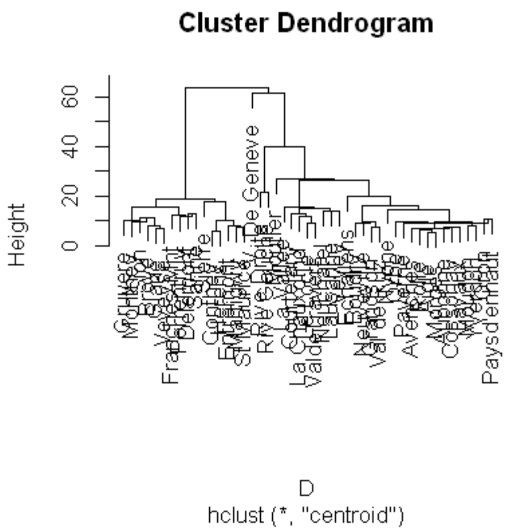
Start to analyse from the right of the curve (curve going from k=N to k = 1). With this curve we would surely choose k = 4

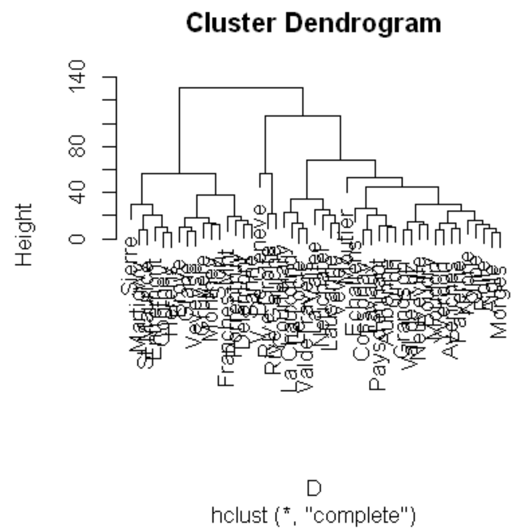
2.0.1 Exercice : Run hcust with the different distance, display and show the results.

"ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).



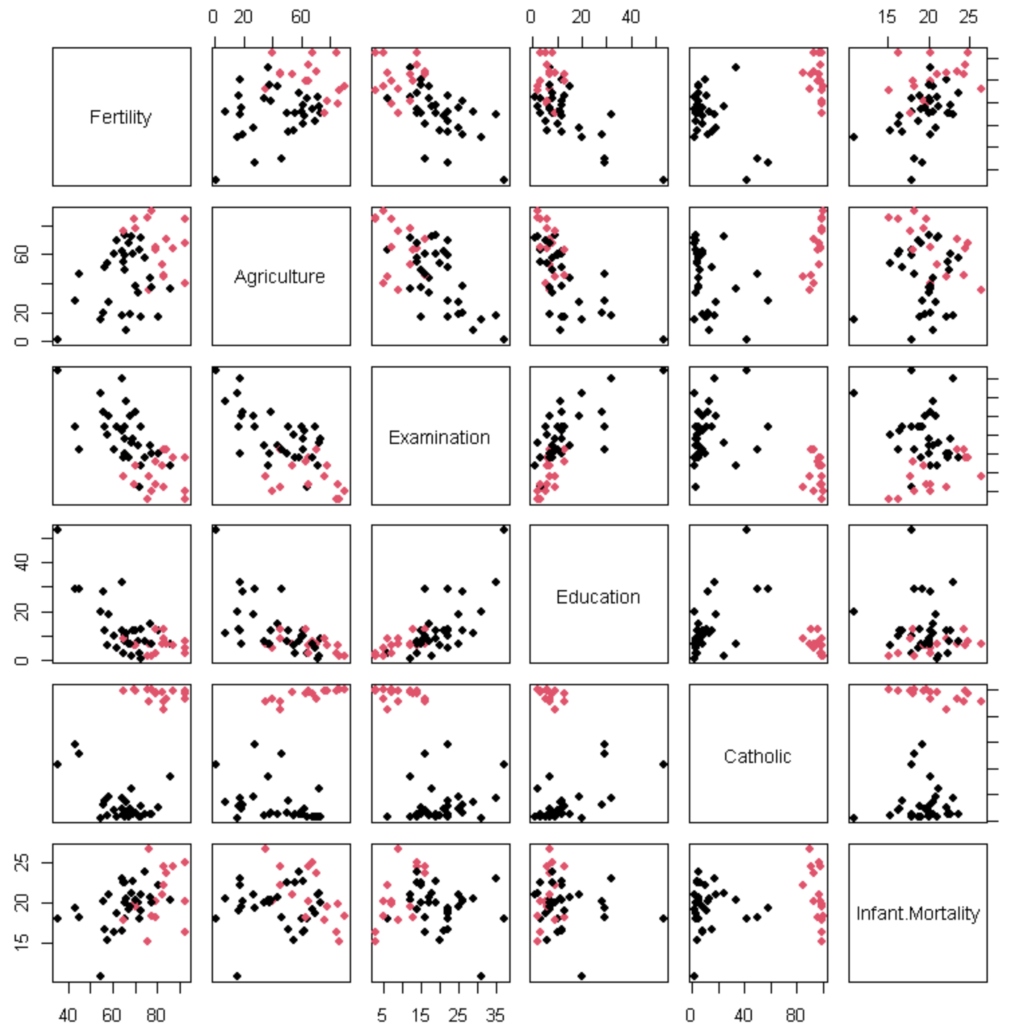




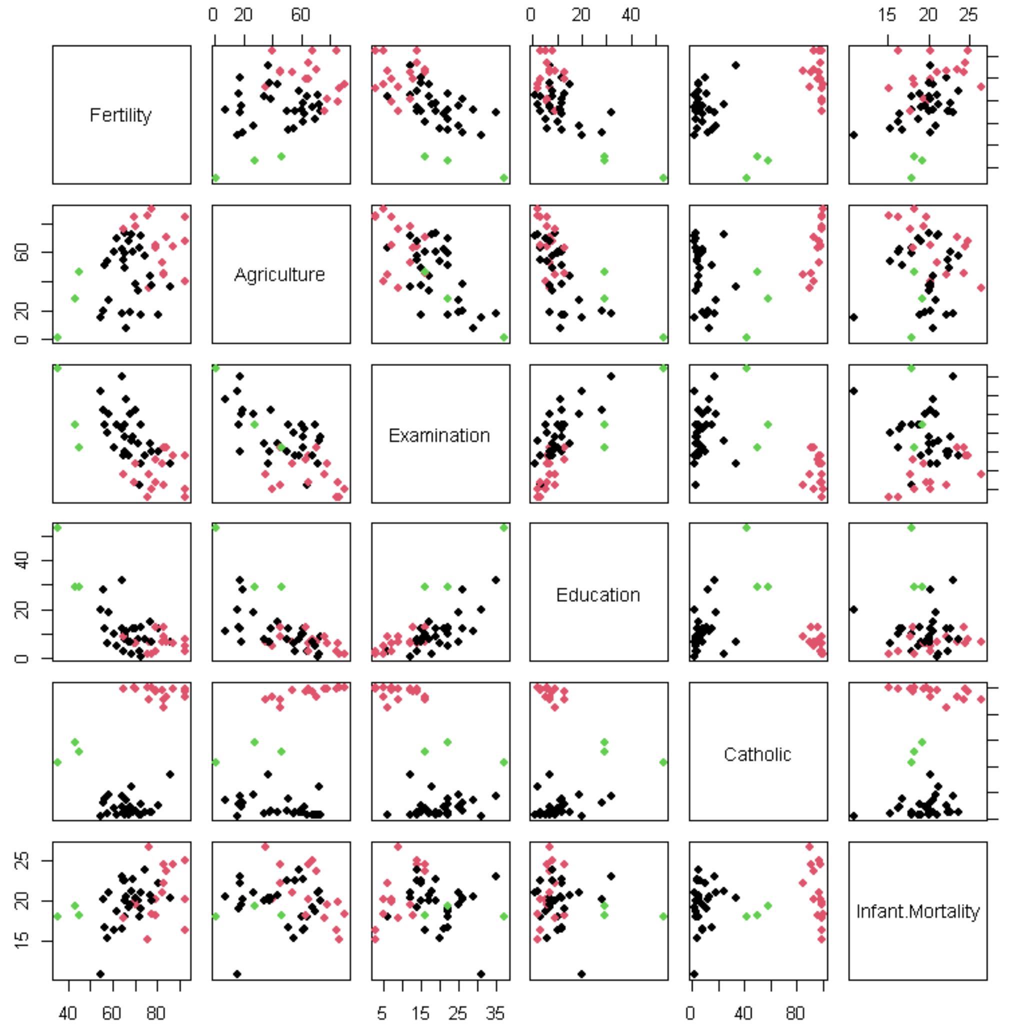


- if looking like a stair("single","centroid"), the method is failing
- "complete" and "ward" are ok for the analysis
- we would choose k=2 for "ward" and k = 3 for "complete"

We cut the trees and look at the results using pairs plot in order to understand the data

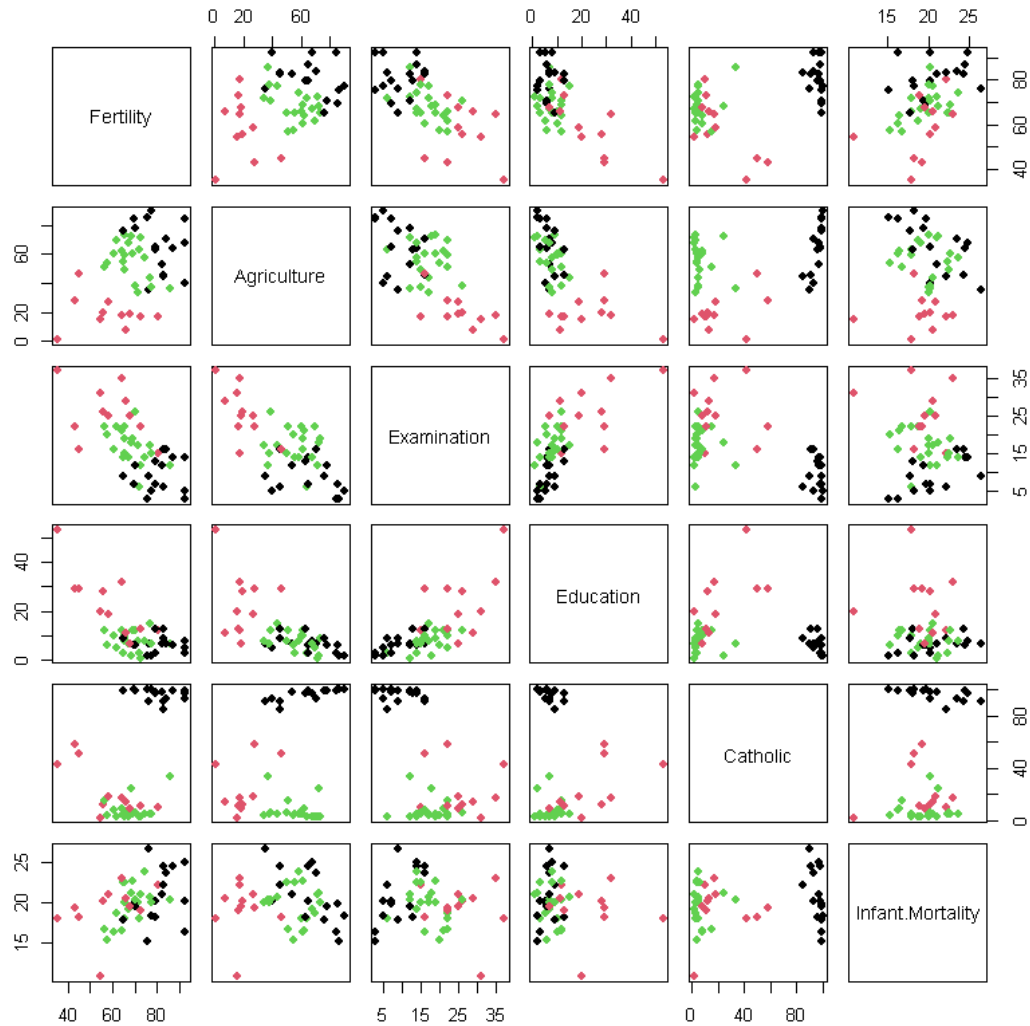


Fertility
Agriculture
Examination
Education
Catholic
Infant.Mortality



With 3 groups the green points are well detached to other (for education, fertility) so the interpretation is really more interesting then with k=2 : better understanding of the datas

2.1 Compare the results of K-Means and HCA both with K=3



Better clustering for HClust, but remember that for K-Means the best k was 4

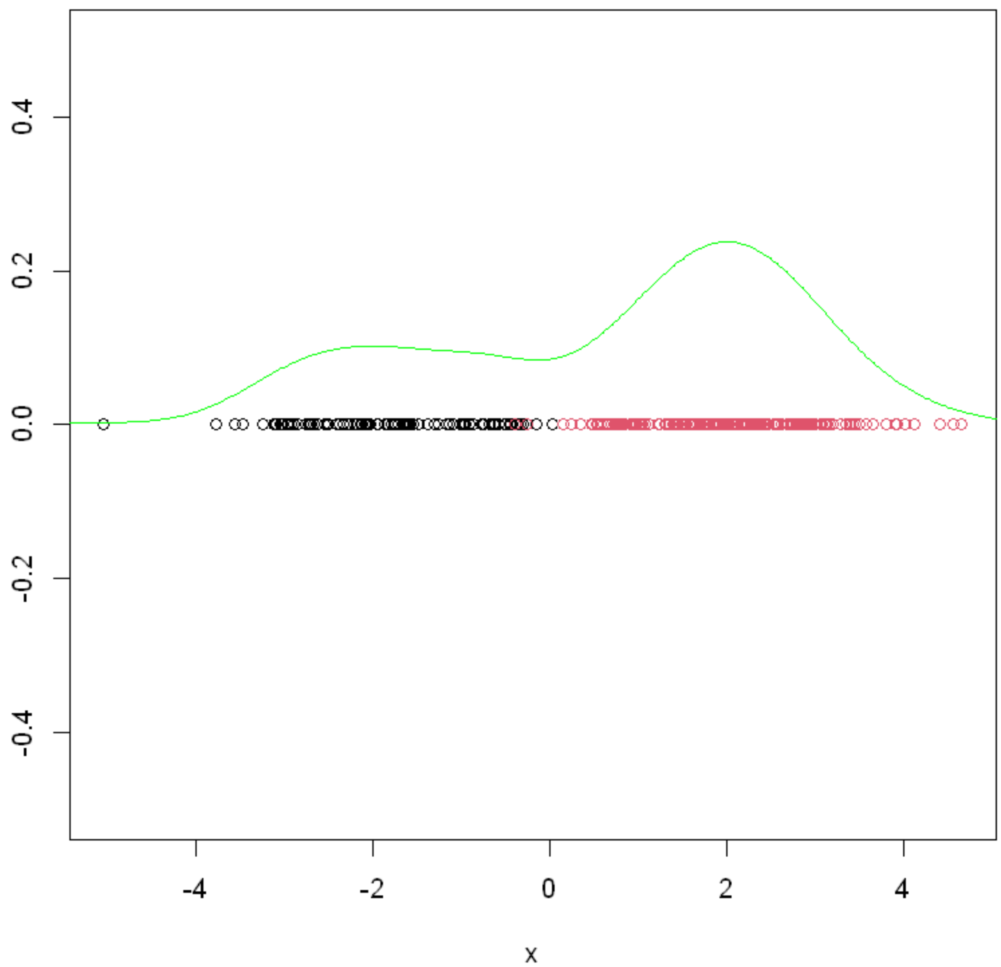
In conclusion : * k= 3 HCA is better * k = 4 K-Means is interesting because the protestant group is divided in an interesting way

3 EM Algorythm with Gaussian Mixture Model (GMM)

3.1 Try to code simple EM algorythm ofr GMM with fixed covariances matrices to simplify

identity matrix

We simulate some data



This is P(x)

View the effect of each iteration of the EM on the mean, the proportions and the cluster menbership

3.1.1 With swiss data and Mclust package

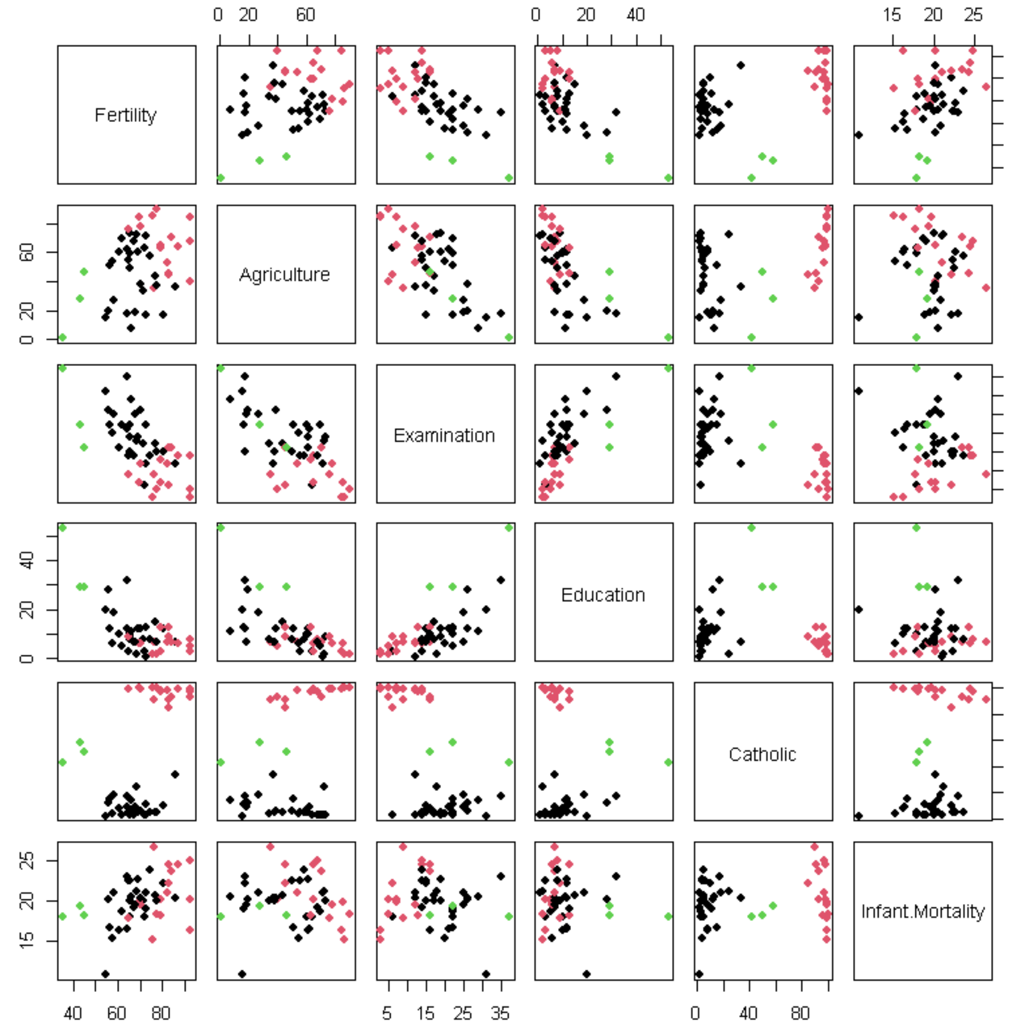
```
Warning message:
"package 'mclust' is in use and will not be installed"
```

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 3
components:
```

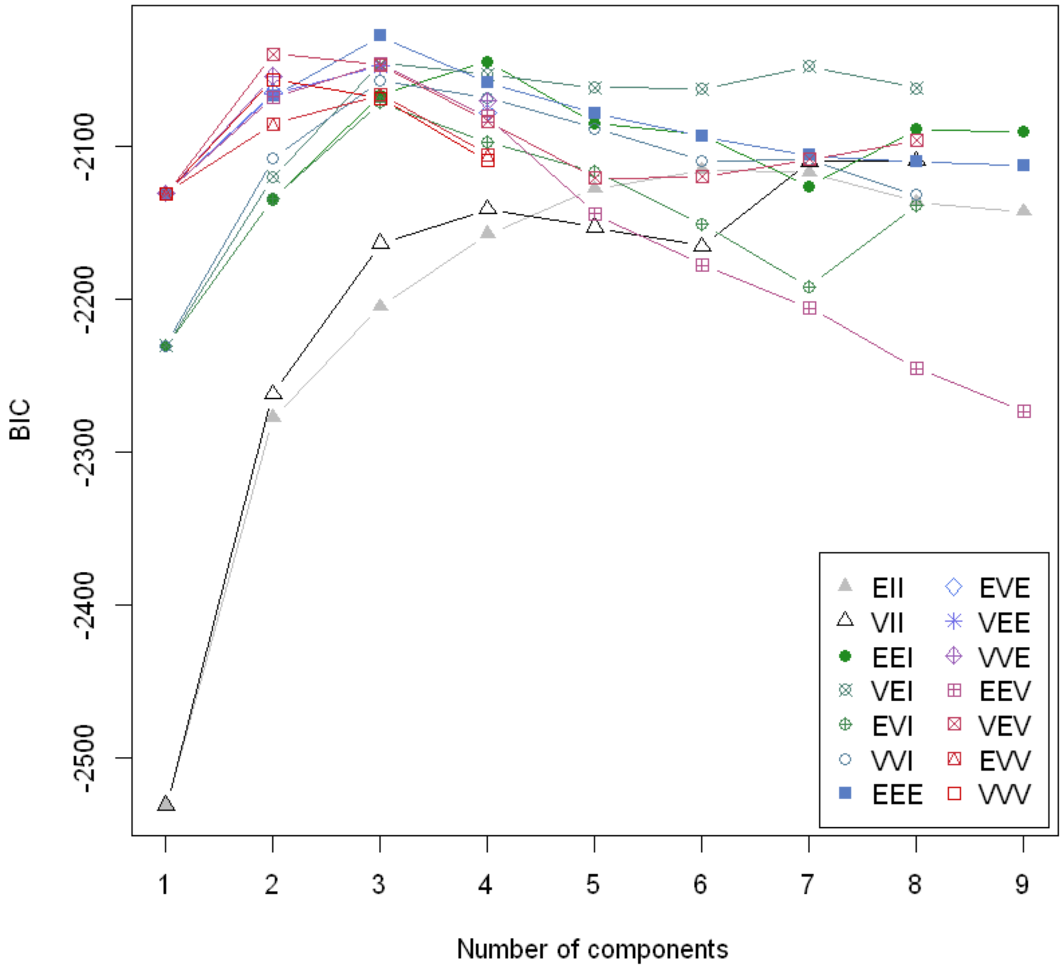
```
 log-likelihood  n df       BIC       ICL
      -934.9916 47 41 -2027.839 -2027.839

Clustering table:
 1  2  3
28 16  3
```

1. 'call' 2. 'data' 3. 'modelName' 4. 'n' 5. 'd' 6. 'G' 7. 'BIC' 8. 'loglik' 9. 'df' 10. 'bic' 11. 'icl' 12. 'hypvol' 13. 'parameters' 14. 'z' 15. 'classification' 16. 'uncertainty'

Courtelary 1 **Delemont** 2 **Franches-Mnt** 2 **Moutier** 1 **Neuveville** 1 **Porrentruy** 2 **Broye** 2 **Glane** 2 **Gruyere** 2 **Sarine** 2 **Veveyse** 2 **Aigle** 1 **Aubonne** 1 **Avenches** 1 **Cossonay** 1 **Echallens** 1 **Grandson** 1 **Lausanne** 1 **La Vallee** 1 **Lavaux** 1 **Morges** 1 **Moudon** 1 **Nyone** 1 **Orbe** 1 **Oron** 1 **Payerne** 1 **Paysd'enhaut** 1 **Rolle** 1 **Vevey** 1 **Yverdon** 1 **Conthey** 2 **Entremont** 2 **Herens** 2 **Martigwy** 2 **Monthey** 2 **St Maurice** 2 **Sierre** 2 **Sion** 2 **Boudry** 1 **La Chauxdfnd** 1 **Le Locle** 1 **Neuchatel** 1 **Val de Ruz** 1 **ValdeTravers** 1 **V. De Geneve** 3 **Rive Droite** 3 **Rive Gauche** 3

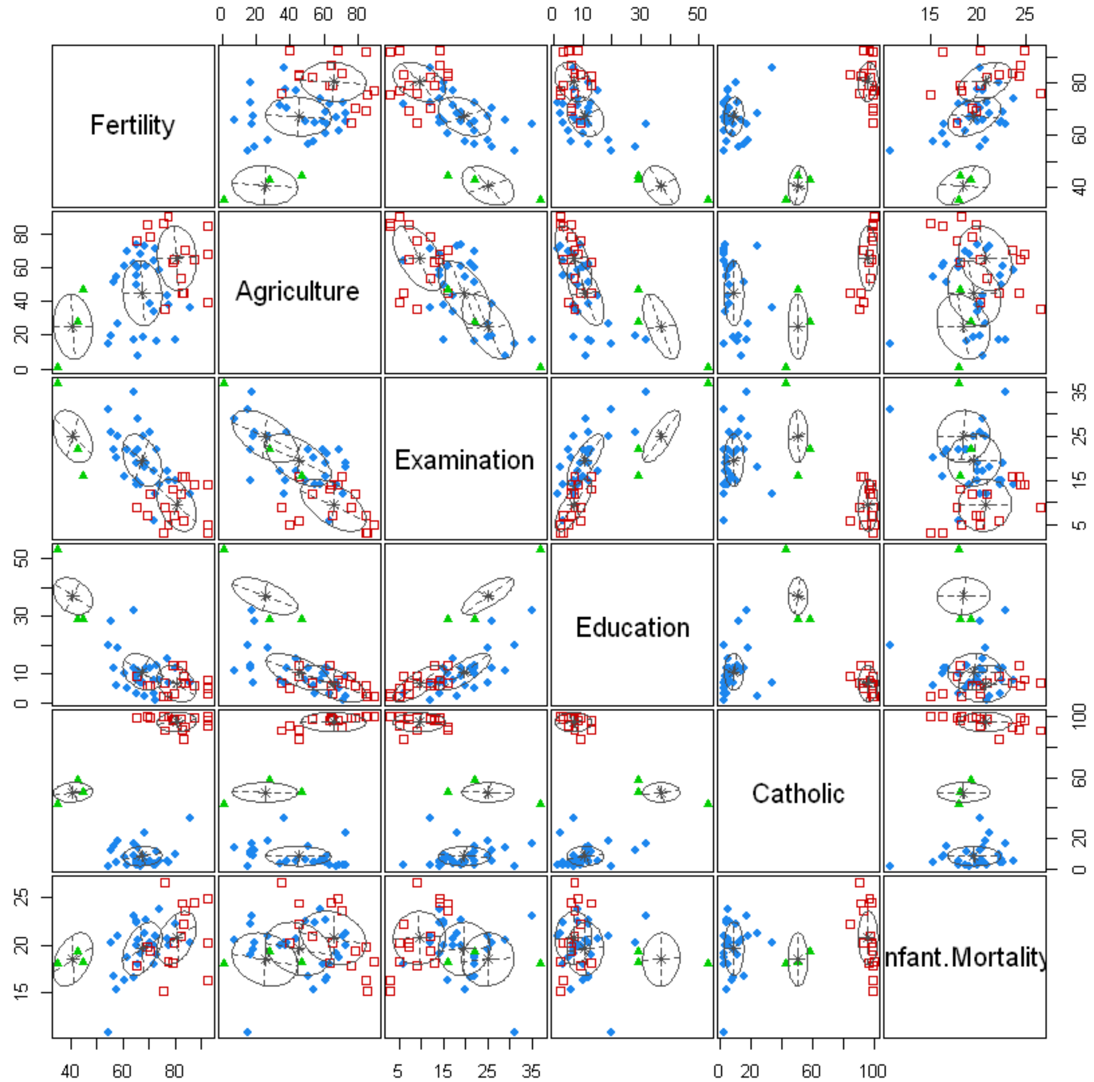


With default parameter, very close to result of HAC with complete linkage



How to to understand this plot : * the EEE model : E = equal, EEE : proportion equal, mean , variance * best of the best : EEE for 3 component * 14 models * 9 values of K tested ! * pos 1 : proportion : E : Equal, V = Free * pos 2 : covariance matrix : V : free, I = identity, E : equal * pos 3 : ??

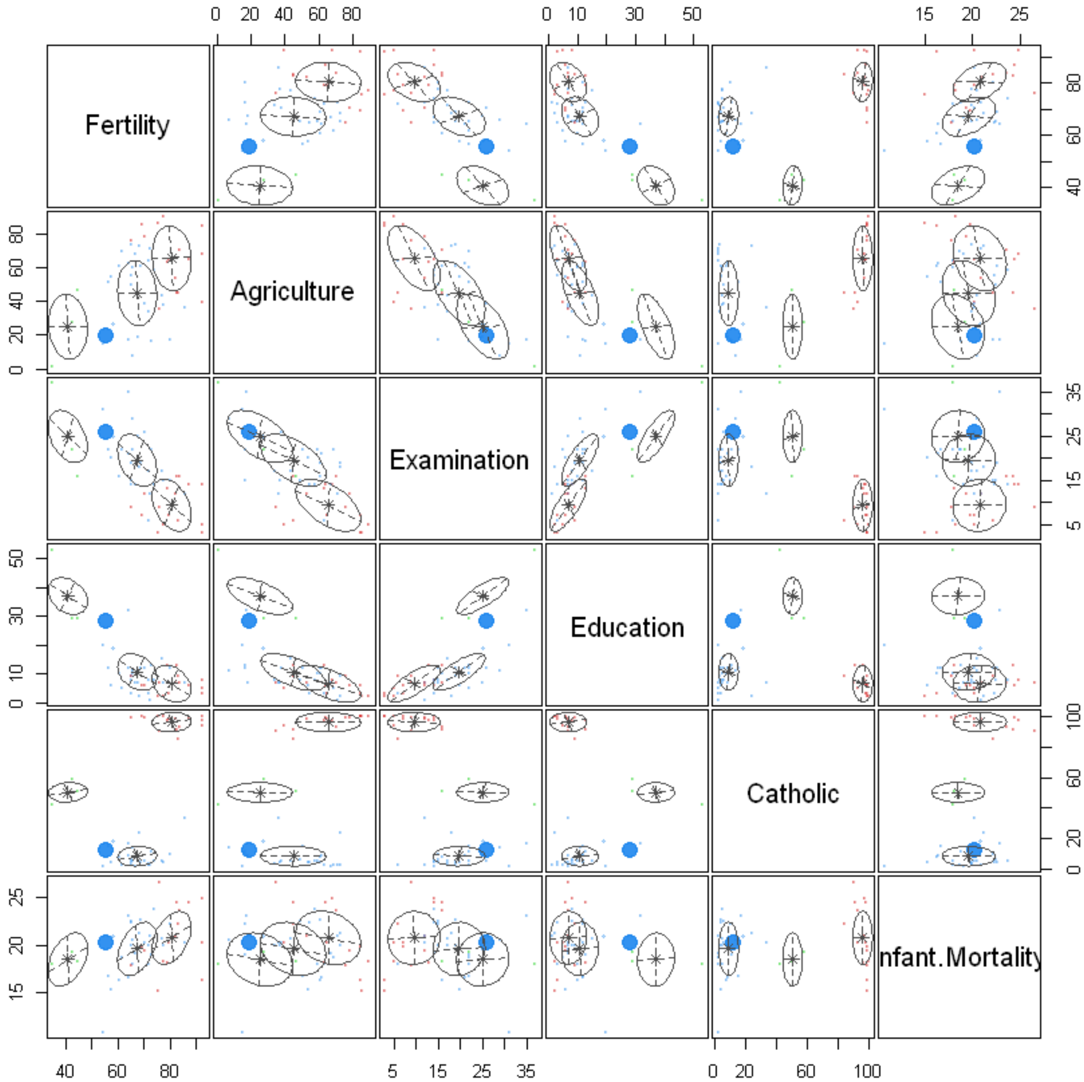
If you are expert, it can be intersting to change k and find the best model with a different k



We can also visualize the estimated Gaussian with the previous graph

Access also to the probability

The bigger the point is the bigger the uncertainty to belong to the choosen cluster is.



This information is direcly extract from the output of the Algorythm.

A matrix: 47 × 3 of type dbl

Courtelary	1.000000e+00	1.936575e-41	7.572986e-21
Delemont	4.918818e-33	1.000000e+00	3.020169e-14
Franches-Mnt	4.930018e-40	1.000000e+00	4.121262e-22
Moutier	1.000000e+00	5.704706e-19	1.283381e-12
Neuveville	1.000000e+00	2.002089e-45	2.831415e-22
Porrentruy	6.729864e-38	1.000000e+00	1.657370e-15
Broye	1.373874e-38	1.000000e+00	2.197023e-22
Glane	3.066321e-43	1.000000e+00	1.218175e-25
Gruyere	2.375623e-43	1.000000e+00	9.018504e-23
Sarine	1.099925e-37	1.000000e+00	2.392042e-18
Veveyse	2.954360e-44	1.000000e+00	2.005166e-25
Aigle	1.000000e+00	1.457455e-43	1.282937e-21
Aubonne	1.000000e+00	4.430158e-48	1.016653e-23
Avenches	1.000000e+00	3.607653e-46	2.536988e-22
Cossonay	1.000000e+00	8.688952e-50	2.364852e-27
Echallens	1.000000e+00	7.755727e-29	1.065739e-17
Grandson	1.000000e+00	1.478713e-48	2.309841e-25
Lausanne	1.000000e+00	3.814979e-39	3.869423e-10
La Vallee	1.000000e+00	1.470315e-52	4.020816e-24
Lavaux	1.000000e+00	3.806609e-48	2.532751e-24
Morges	1.000000e+00	4.013728e-47	7.548107e-25
Moudon	1.000000e+00	1.991294e-46	3.822450e-23
Nyone	1.000000e+00	1.179617e-37	2.005990e-16
Orbe	1.000000e+00	1.321625e-48	1.535777e-24
Oron	1.000000e+00	3.224140e-48	5.153818e-27
Payerne	1.000000e+00	7.294973e-45	1.117981e-22
Paysd'enhaut	1.000000e+00	7.474317e-47	3.158982e-23
Rolle	1.000000e+00	2.267946e-43	2.938428e-19
Vevey	1.000000e+00	3.564403e-34	6.222807e-12
Yverdon	1.000000e+00	1.696589e-44	2.529561e-20
Conthey	7.303432e-47	1.000000e+00	1.053782e-22
Entremont	2.668420e-47	1.000000e+00	5.984137e-20
Herens	3.734582e-47	1.000000e+00	1.192764e-23
Martigwy	4.178559e-45	1.000000e+00	2.812089e-21
Monthey	1.023760e-44	1.000000e+00	1.288274e-22
St Maurice	2.720172e-46	1.000000e+00	7.009389e-18
Sierre	1.557015e-46	1.000000e+00	6.786755e-27
Sion	3.215546e-43	1.000000e+00	2.654542e-20
Boudry	1.000000e+00	9.048151e-48	1.521805e-26
La Chauxdfnd	1.000000e+00	1.195468e-41	5.775682e-23
Le Locle	1.000000e+00	3.803771e-42	4.647618e-22
Neuchatel	1.000000e+00	4.741972e-36	1.602488e-12
Val de Ruz	1.000000e+00	1.077673e-46	7.180907e-26
ValdeTravers	1.000000e+00	4.633954e-46	2.245377e-26
V. De Geneve	1.931260e-19	2.996057e-28	1.000000e+00
Rive Droite	8.232250e-19	9.681768e-18	1.000000e+00
Rive Gauche	1.927605e-20	1.019775e-13	1.000000e+00

Other package implement the EM algorithm for continuous but also categorical one. Rmixmod package allows to deal with both.

It is also possible to get the datas of the 'average guy' of each cluster

A matrix: 6 × 3 of type dbl

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

Ex : cluster the wine data and evaluate the quality of the clustering regarding the know labels

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust EVI (diagonal, equal volume, varying shape) model with 3 components:

 log-likelihood   n  df       BIC       ICL
      -11557.21 178 162 -23953.87 -23955.04

Clustering table:
 1  2  3
65 63 50
```

1. 'call' 2. 'data' 3. 'modelName' 4. 'n' 5. 'd' 6. 'G' 7. 'BIC' 8. 'loglik' 9. 'df' 10. 'bic' 11. 'icl' 12. 'hypvol' 13. 'parameters' 14. 'z' 15. 'classification' 16. 'uncertainty'

Mclust find by itself the three clusters

A way to compare : confusion matrix

```
             1  2  3
  Barbera     0  0 48
  Barolo     58  1  0
  Grignolino  7 62  2
```

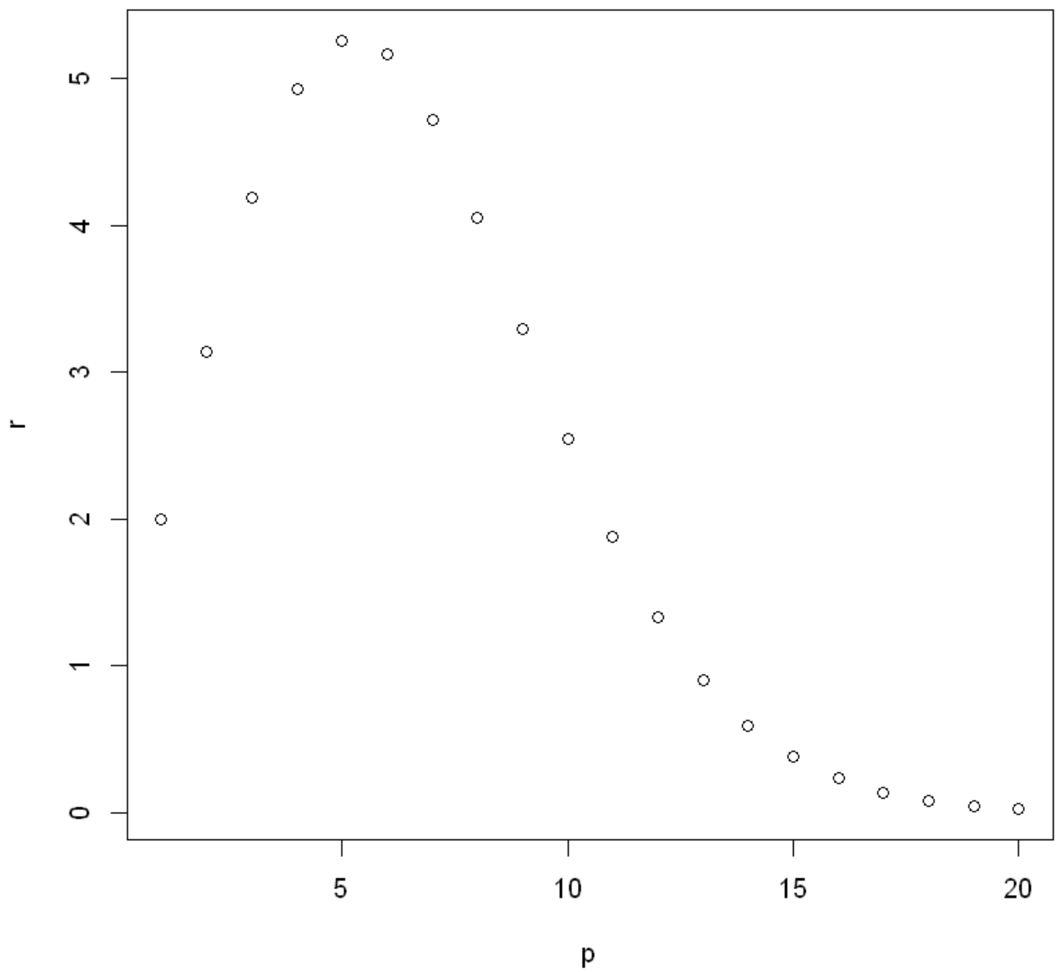
Globaly the result is very good : error = 10 errors over 178 = 5%

Try the same classification with K-Means

1. 'cluster' 2. 'centers' 3. 'totss' 4. 'withinss' 5. 'tot.withinss' 6. 'betweenss' 7. 'size' 8. 'iter' 9. 'ifault'

```
             1  2  3
  Barbera     6  0 42
  Barolo     32 24  3
  Grignolino  6  0 65
```

With Kmeans the error ratio is 25% (5% with Hclust)



4 Dimension Reduction with PCA

```
Warning message:
"package 'FactoMineR' is in use and will not be installed"
```

A data.frame: 6 × 13

	100m	Long.jump	Shot.put	High.jump	400m	110m.hurdle	Discus
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
SEBRLE	11.04	7.58	14.83	2.07	49.81	14.69	43.75
CLAY	10.76	7.40	14.26	1.86	49.37	14.05	50.72
KARPOV	11.02	7.30	14.77	2.04	48.37	14.09	48.95
BERNARD	11.02	7.23	14.25	1.92	48.93	14.99	40.87
YURKOV	11.34	7.09	15.19	2.10	50.42	15.31	46.26
WARNERS	11.11	7.60	14.31	1.98	48.68	14.23	41.10

We just take the 10 first columns

Run PCA on those datas

```
Call:
princomp(x = X)

Standard deviations:
     Comp.1      Comp.2      Comp.3      Comp.4      Comp.5      Comp.6
11.61065403  4.78910847  3.12206072  1.05698409  0.58972067  0.36425523
     Comp.7      Comp.8      Comp.9     Comp.10
 0.24917123  0.22222732  0.15825005  0.07006272

 10  variables and  41 observations.
```

Now we have to select the number of dimension component to retain

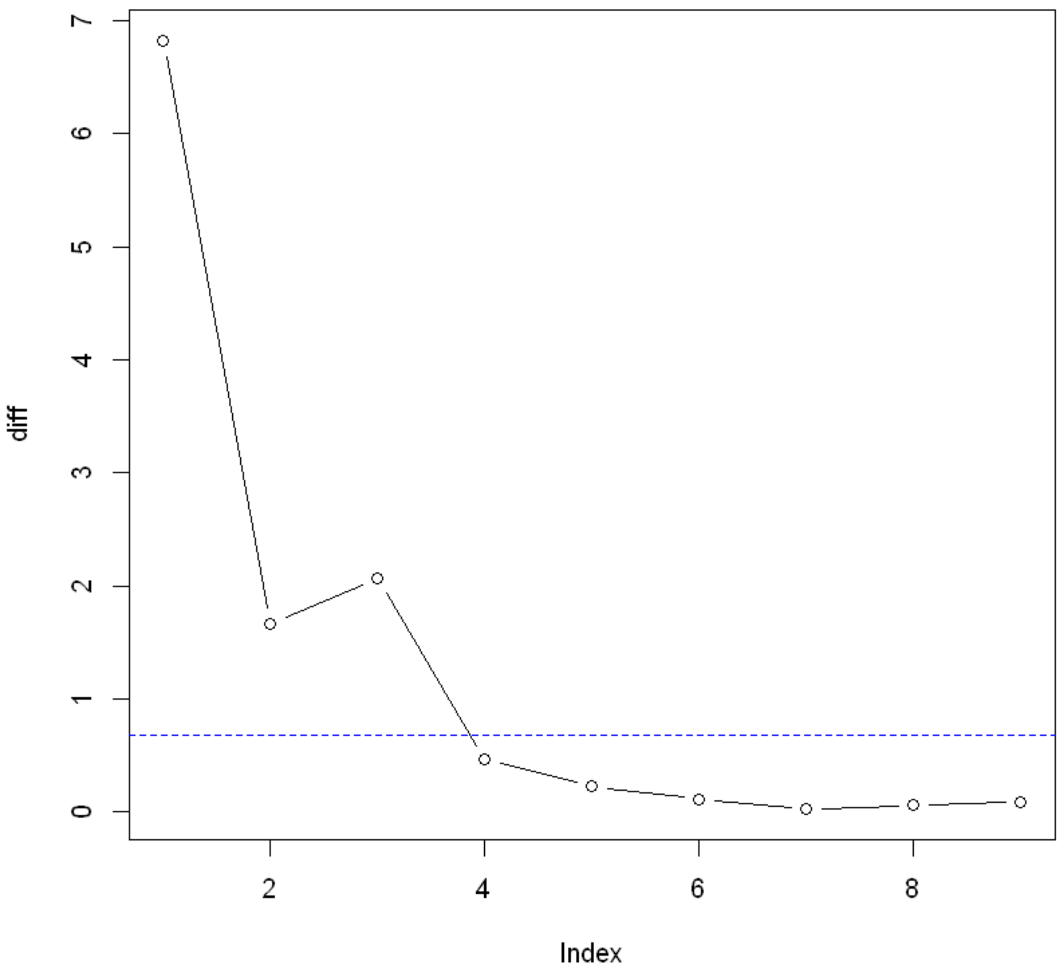
1) look at the summary and use the cumulative proportion regarding 90% => we should here keep 2 components (93,2%)

```
Importance of components:
                           Comp.1    Comp.2    Comp.3      Comp.4      Comp.5
Standard deviation     11.6106540 4.7891085 3.1220607 1.056984087 0.589720672
Proportion of Variance  0.7965959 0.1355296 0.0575980 0.006601788 0.002055026
Cumulative Proportion   0.7965959 0.9321255 0.9897235 0.996325247 0.998380274
                             Comp.6      Comp.7      Comp.8       Comp.9
Standard deviation     0.3642552340 0.249171229 0.222227318 0.1582500495
Proportion of Variance 0.0007840365 0.000366877 0.000291823 0.0001479832
Cumulative Proportion  0.9991643100 0.999531187 0.999823010 0.9999709933
                            Comp.10
Standard deviation     7.006272e-02
Proportion of Variance 2.900673e-05
Cumulative Proportion  1.000000e+00
```

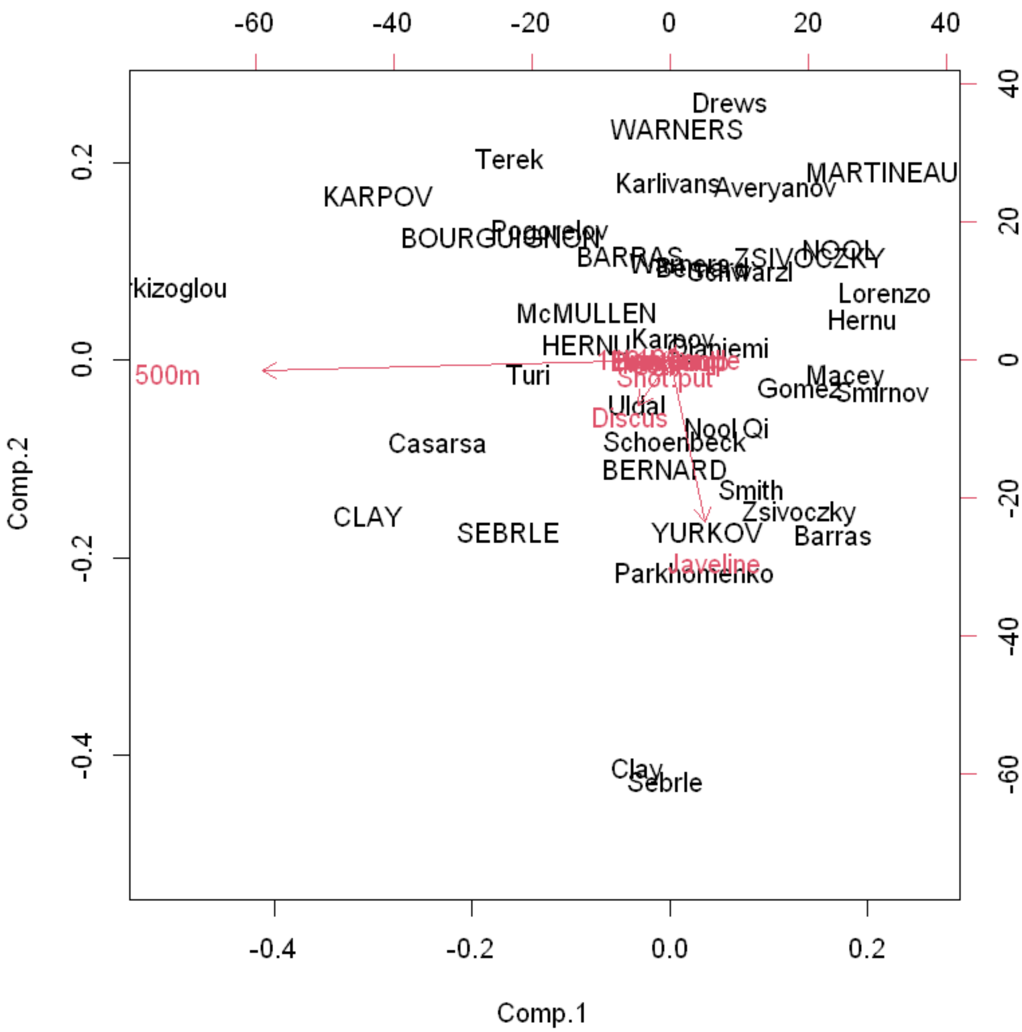
2) use the screeplot Applying the rule of the break, the choice could be 3.

screeplot(pc)

3) with scree-test of Cattel, the choice is also d= 3



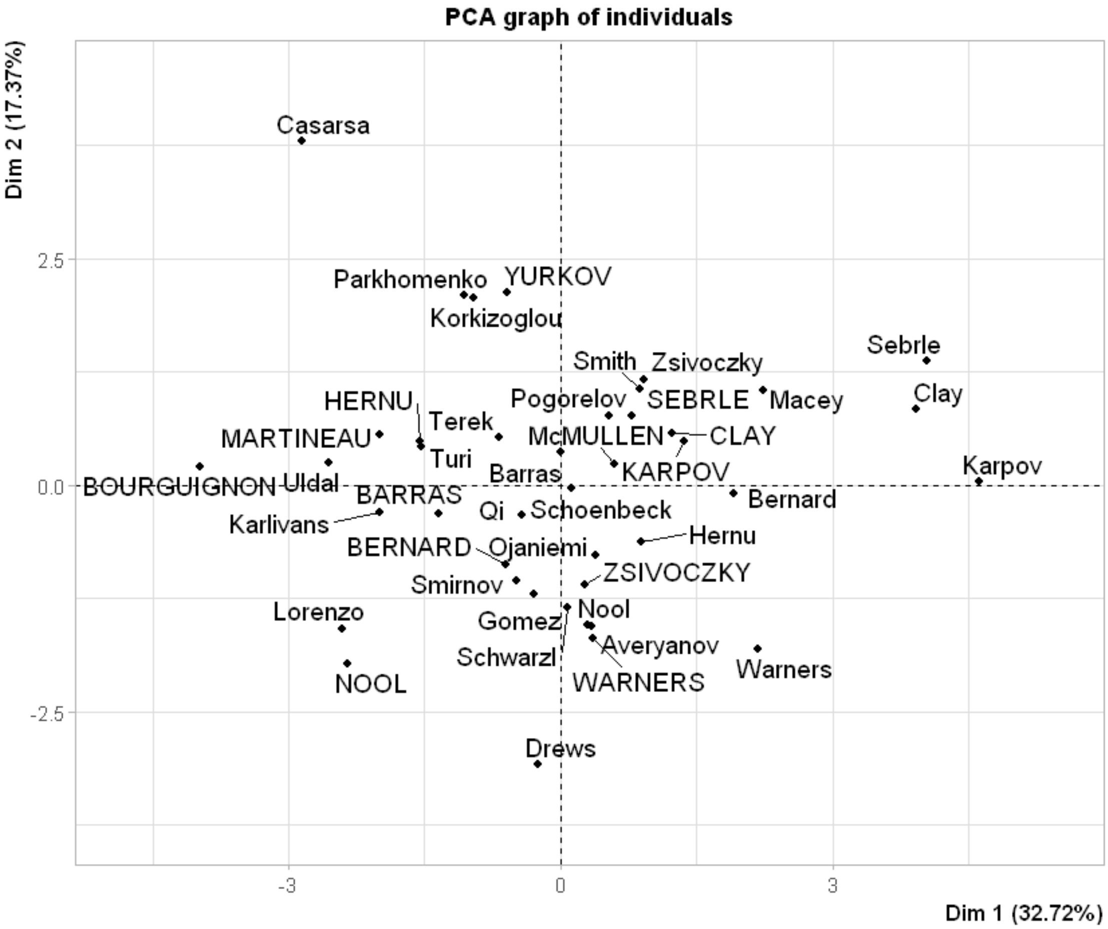
We can now look art the correlation circle

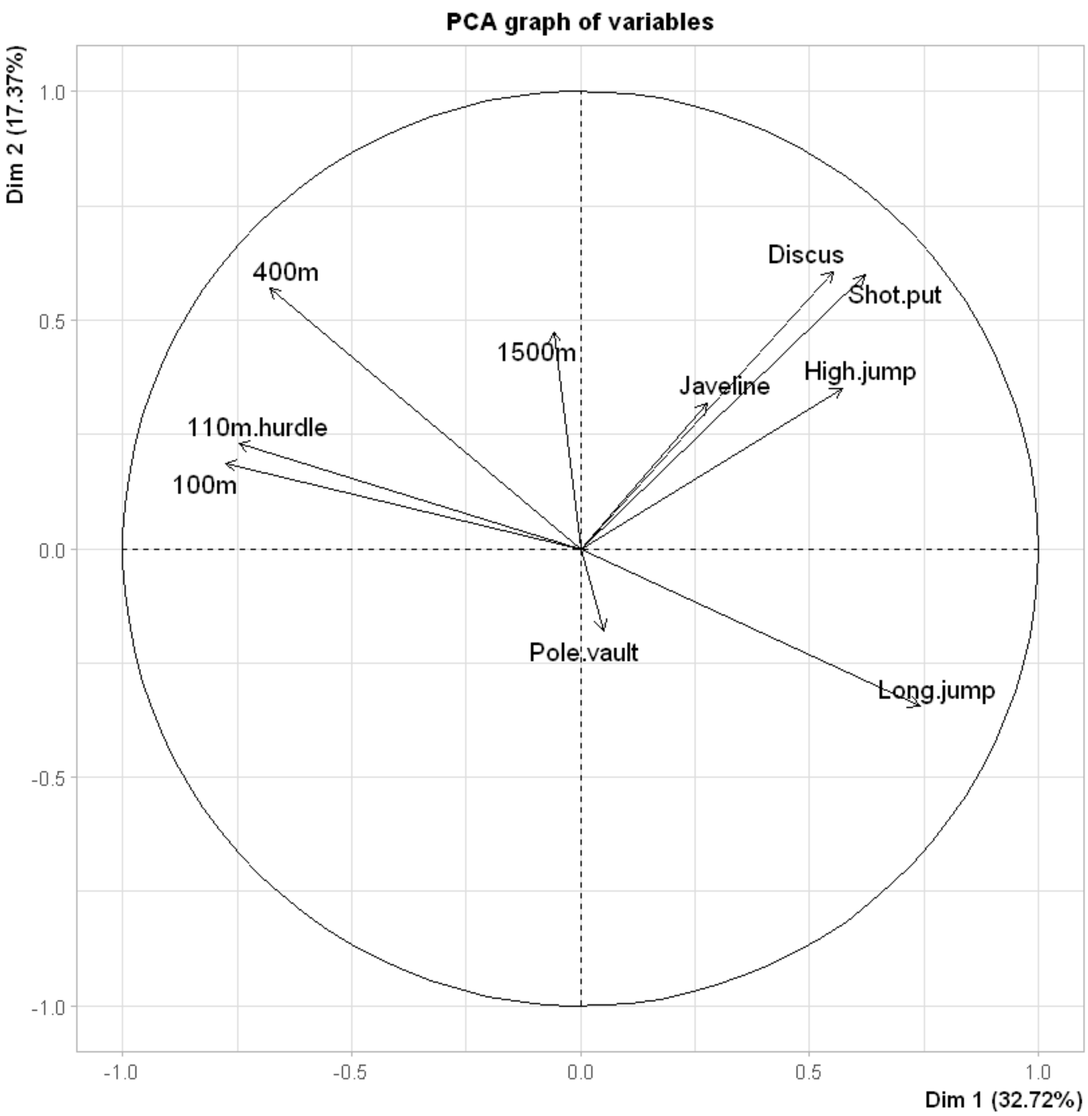


in red : correlation circle

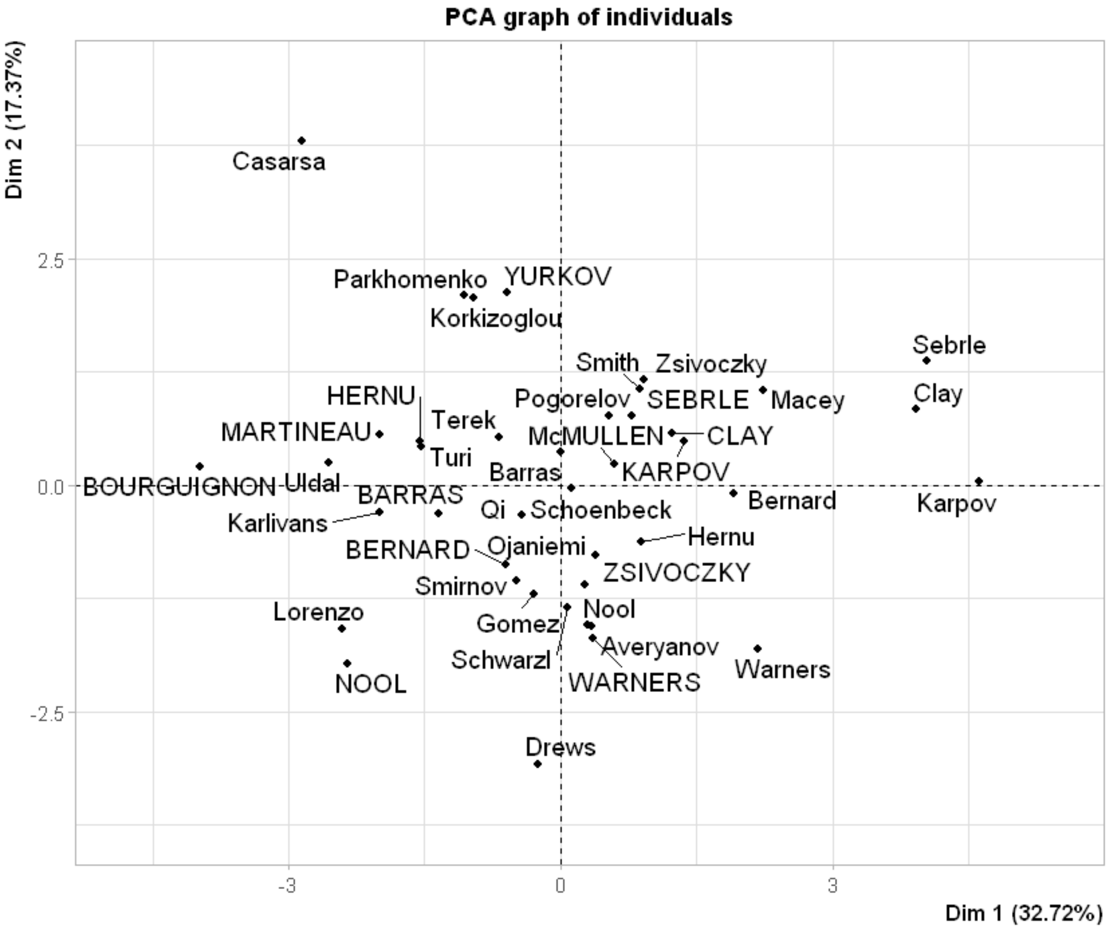
in black : score plot : projection of the datas on the two best coponent

To have a clearer view of correlation circle, it’s better to use (FactoMineR package)


PCA graph of individuals
Dim 2 (17.37%)
Dim 1 (32.72%)
2.5
0.0
-2.5
-3
0
3
Casarsa
Parkhomenko
YURKOV
Korkizoglou
Smith
Zsivoczky
Sebrle
Clay
Karpov
HERNU
Terek
Pogorelov
SEBRLE
Macey
MARTINEAU
McMULLEN
CLAY
Turi
KARPOV
Barras
BOURGUIGNON
Uldal
BARRAS
Bernard
Karlivans
Qi
Schoenbeck
Hernu
BERNARD
Ojaniemi
ZSIVOCZKY
Smirnov
Lorenzo
Gomez
Nool
Averyanov
Schwarzl
Warners
NOOL
WARNERS
Drews



PCA graph of variables
Dim 2 (17.37%)
Dim 1 (32.72%)
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
Discus
400m
Shot.put
1500m
Javeline
High.jump
110m.hurdle
100m
Pole.vault
Long.jump



5 Clustering with HDclassif

```
Warning message:
"package 'HDclassif' is in use and will not be installed"
```

```
Y             1  2  3
  Barbera     0  0 48
  Barolo     58  1  0
  Grignolino  2 68  1
```

With this HDCC algorythme the result is even better then with : 2.2% of error