Лабораторна робота №5 Кластеризація

студентки групи ЕП-61

Лузанової О.С.

Мета роботи – провести процедуру кластеризаціїї різними методами, а саме: Ієрархічна кластеризація, Кластеризація на основі k-means, за допомогою карти Кохонена

Після виконання роботи студент повинен:

ЗНАТИ сутність кластеризації

УМІТИ організувати та провести зібр даних, кодування, обчислення та інтерпретування статистик, видалити викиди та провести шкалювання даних, обробити відсутні значення – як етап підготовки даних для кластеризації та безпосередньо саму кластеризацію.

**Кластеризація** – групування об’єктів на основі їх властивостей. Об’єкти в середині кластера повинні бути схожими і відрізнятися від об’єктів, що увійшли в інші кластери. Чим більше схожі об’єкти в середині кластера і чим більше відмінностей між кластерами, тим точніше кластеризація.

Для виконання Лабораторної роботи №5 було обрано тему – збір даних та класифікація рівня злочинності та економічних факторів. Дані зібрані по країнах за 2017 рік.

Щоб виявити економічні фактори, що впливають на кількість вбивст, скоєних у країні, було проведено аналіз наукової літератури. Обрано такі показники, як:

1. Кількість населення (Population).
2. ВВП на душу населення у доларах США (GDP\_p)
3. Рівень безробіття (Unem).
4. Ймовірна тривалість життя (Life\_ex).
5. Рівень смертності (Death).
6. Рівень народжувальності (Bir).
7. % зайнятих віком від 15 років (Emp).
8. % зайнятих у сфері послуг (Emp\_.s)
9. % зайнятих у сфері с/г (Emp\_a).
10. % зайнятих у сфері виробництва (Emp\_i).
11. Державні видатки у сферу освіти (edu\_ex).
12. Державні видатки у сферу збройних сил (Mil\_ex).

lab5\_luzanova\_CLUSTERIZATION

# Hierarchical clustering

# Download the data

library (psych)

## Warning: package 'psych' was built under R version 3.6.3

set.seed(123)  
f <- read.csv2('crime\_index\_pre.csv', header = TRUE, encoding = 'UNICOD')  
f= subset(f, select = -c(1))  
head(f)

## cr\_in Population gdp\_p unem life\_ex dea bir emp emp\_s emp\_a  
## 1 1 36296400 1393.233 11.18 64.130 6.58 33.167 43.220 37.99 43.990  
## 2 0 2873457 4531.020 13.75 78.330 7.71 11.930 48.220 42.44 38.200  
## 3 1 41389198 4044.280 12.00 76.500 4.72 24.850 39.516 58.85 10.160  
## 4 1 29816748 4095.810 7.12 63.064 8.43 33.167 72.050 41.07 50.560  
## 5 1 44044811 14591.860 8.35 76.370 7.62 17.210 55.530 77.50 1.115  
## 6 0 2944809 3914.500 17.68 74.800 9.86 14.300 45.800 51.95 31.340  
## emp\_i edu\_ex mil\_ex  
## 1 18.030 530307037 191407113  
## 2 19.350 431032412 144382689  
## 3 30.990 7361823737 10073364021  
## 4 8.517 4091869979 3062872914  
## 5 22.440 31276833378 5459643672  
## 6 16.710 386855546 443610413

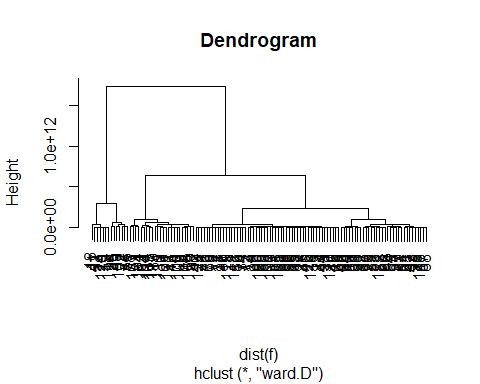
describe(f)

## vars n mean sd median trimmed  
## cr\_in 1 123 5.000000e-01 5.000000e-01 0.000000e+00 4.900000e-01  
## Population 2 123 3.460812e+07 4.958875e+07 1.075468e+07 2.292767e+07  
## gdp\_p 3 123 1.813714e+04 1.821617e+04 1.025423e+04 1.528928e+04  
## unem 4 123 7.300000e+00 4.580000e+00 5.760000e+00 6.740000e+00  
## life\_ex 5 123 7.518000e+01 5.750000e+00 7.583000e+01 7.570000e+01  
## dea 6 123 7.730000e+00 2.690000e+00 7.110000e+00 7.570000e+00  
## bir 7 123 1.658000e+01 7.440000e+00 1.430000e+01 1.565000e+01  
## emp 8 123 5.689000e+01 1.047000e+01 5.802000e+01 5.661000e+01  
## emp\_s 9 123 6.039000e+01 1.530000e+01 6.216000e+01 6.145000e+01  
## emp\_a 10 123 1.767000e+01 1.738000e+01 1.070000e+01 1.504000e+01  
## emp\_i 11 123 2.147000e+01 6.420000e+00 2.048000e+01 2.153000e+01  
## edu\_ex 12 123 1.901020e+10 3.067557e+10 4.501641e+09 1.132612e+10  
## mil\_ex 13 123 8.330987e+09 1.273161e+10 3.062873e+09 5.176146e+09  
## mad min max range skew kurtosis  
## cr\_in 0.000000e+00 0.00 1.000000e+00 1.000000e+00 0.02 -2.02  
## Population 1.306531e+07 647881.60 1.877530e+08 1.871052e+08 1.99 3.09  
## gdp\_p 1.128785e+04 1393.23 6.081815e+04 5.942492e+04 1.12 -0.02  
## unem 3.850000e+00 1.82 1.768000e+01 1.586000e+01 0.92 -0.18  
## life\_ex 6.000000e+00 63.06 8.266000e+01 1.959000e+01 -0.63 -0.44  
## dea 2.950000e+00 3.49 1.327000e+01 9.780000e+00 0.50 -0.65  
## bir 6.230000e+00 8.30 3.317000e+01 2.487000e+01 0.89 -0.30  
## emp 9.270000e+00 39.52 7.781000e+01 3.829000e+01 0.10 -0.61  
## emp\_s 1.661000e+01 28.61 8.161000e+01 5.299000e+01 -0.47 -0.78  
## emp\_a 1.259000e+01 1.12 5.969000e+01 5.857000e+01 1.04 0.01  
## emp\_i 6.260000e+00 8.52 3.305000e+01 2.454000e+01 -0.02 -0.54  
## edu\_ex 6.003692e+09 386855546.29 1.196822e+11 1.192953e+11 2.22 4.19  
## mil\_ex 4.030544e+09 71955687.86 4.632868e+10 4.625672e+10 2.03 3.19  
## se  
## cr\_in 5.000000e-02  
## Population 4.471267e+06  
## gdp\_p 1.642500e+03  
## unem 4.100000e-01  
## life\_ex 5.200000e-01  
## dea 2.400000e-01  
## bir 6.700000e-01  
## emp 9.400000e-01  
## emp\_s 1.380000e+00  
## emp\_a 1.570000e+00  
## emp\_i 5.800000e-01  
## edu\_ex 2.765923e+09  
## mil\_ex 1.147971e+09

#Висновок: для побудови карт Кохонена використані дані, а саме: індекс злочинності та економічні показники. кількість спостережень – 123, кількість змінних – 13

# Hierarchical clustering

model\_hc <- hclust(dist(f), method = "ward.D" )  
plot(model\_hc, main = paste('Dendrogram'))

 #Висновок: на основі навчальної вибірки побудовано дендрограму з використанням методу Ward.D.

## Fitting HC to the dataset

y\_hc <- cutree(model\_hc, k = 3)  
#cluster cores  
aggregate(f,by=list(y\_hc),FUN=mean)

## Group.1 cr\_in Population gdp\_p unem life\_ex dea bir  
## 1 1 0.5411765 22619274 13550.06 7.217035 74.22566 7.561588 17.64515  
## 2 2 0.5000000 37089296 26737.39 8.299708 76.40300 8.022042 15.54438  
## 3 3 0.2142857 103144134 31244.04 6.090714 78.86886 8.280643 11.89021  
## emp emp\_s emp\_a emp\_i edu\_ex mil\_ex  
## 1 57.53496 57.84872 20.54879 20.96212 3730895783 3467004868  
## 2 54.67746 64.55800 13.08237 22.22042 29305399468 7125433009  
## 3 56.80786 68.71571 8.04000 23.24429 94128529125 39928971198

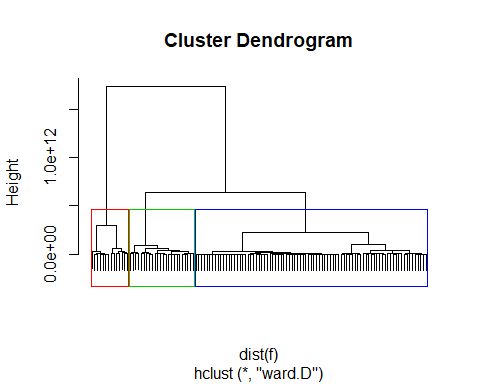
#Cluster stat  
f$hc <- y\_hc  
table(f$hc)

##   
## 1 2 3   
## 85 24 14

#Висновок: на основі аналізу дендрограми виявлено три кластери. Розраховано характеристики типового об’єкту кластерів. До першого кластеру потрапили 85 спостережень, до другого - 24, до третього - 14. До першого, найбільшого кластеру війшли країни з рівнем злочинності вище середнього, малою кількістю населення, малим ВВП на душу населення, з середнім рівнем безробіття, з низкою тривалістю життя, з низьким рівнем смертності, з високим рівнем народжуванності, з рівнем зайнятості вище середнього, з нидьким рівнем зайнятості у сфері послуг та промисловості, з високим рівнем зайнятості у с/г, з низьким рівнем видатків у сферу освіти та збройних сил

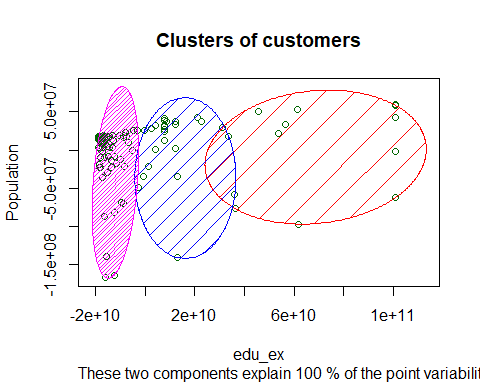
## Plotting the dendrogram

plot(model\_hc, cex = 0.7, labels = FALSE)  
rect.hclust(model\_hc, k = 3, border = 2:5)



## Visualising the clusters

library(cluster)  
clusplot(f[,c('edu\_ex','Population')],  
 y\_hc,  
 lines = 0,  
 shade = TRUE,  
 color = TRUE,  
 labels= 0,  
 plotchar = FALSE,  
 span = TRUE,  
 main = paste('Clusters of customers'),  
 xlab = 'edu\_ex',  
 ylab = 'Population')

 #бачимо, що всі три кластери добре поділені

y\_hc <- cutree(model\_hc, k = 3)  
#cluster cores  
aggregate(f,by=list(y\_hc),FUN=mean)

## Group.1 cr\_in Population gdp\_p unem life\_ex dea bir  
## 1 1 0.5411765 22619274 13550.06 7.217035 74.22566 7.561588 17.64515  
## 2 2 0.5000000 37089296 26737.39 8.299708 76.40300 8.022042 15.54438  
## 3 3 0.2142857 103144134 31244.04 6.090714 78.86886 8.280643 11.89021  
## emp emp\_s emp\_a emp\_i edu\_ex mil\_ex hc  
## 1 57.53496 57.84872 20.54879 20.96212 3730895783 3467004868 1  
## 2 54.67746 64.55800 13.08237 22.22042 29305399468 7125433009 2  
## 3 56.80786 68.71571 8.04000 23.24429 94128529125 39928971198 3

f$hc <- y\_hc  
table(f$hc)

##   
## 1 2 3   
## 85 24 14

f= subset(f, select = -c(hc)) #видалимо стовпець hc

#“K-Means”

## Historgram for each attribute

library(tidyr)  
library(ggplot2)

##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

f %>%   
 gather(Attributes, value, 1:13) %>%   
 ggplot(aes(x=value)) +  
 geom\_histogram(fill = "lightblue2", color = "black") +   
 facet\_wrap(~Attributes, scales = "free\_x") +  
 labs(x = "Value", y = "Frequency")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



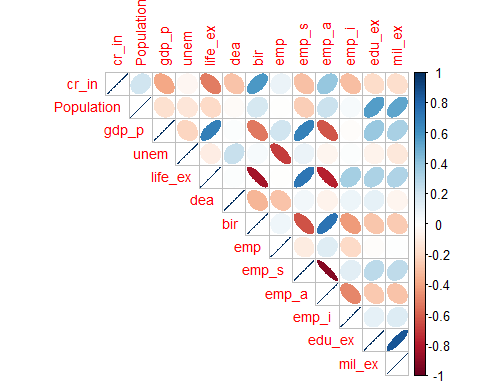
## Correlation

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.6.3

## corrplot 0.84 loaded

corrplot(cor(f), type = "upper", method = "ellipse", tl.cex = 0.9)

 #Бачимо, що змінні між собою корелюють у допустимих межах (низький та нижче середнього рівня), але й такі, що мають високу кореляцію, а саме:life\_ex/bir, life\_ex/emp\_a, emp\_s/emp\_a, edu\_ex/mil\_ex ## Clustree

library(clustree)

## Warning: package 'clustree' was built under R version 3.6.3

## Loading required package: ggraph

## Warning: package 'ggraph' was built under R version 3.6.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.6.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

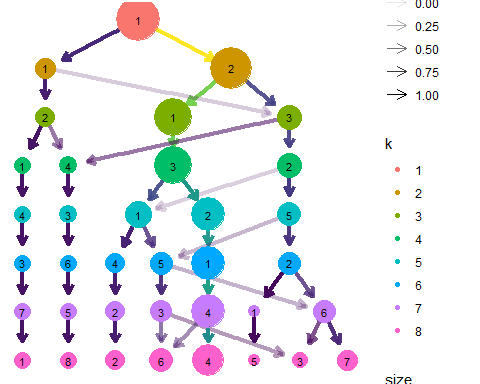
tmp <- NULL  
for (k in 1:8){  
 tmp[k] <- kmeans(f, k, nstart = 30)  
}

## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены

## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены  
  
## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены  
  
## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены  
  
## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены  
  
## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены  
  
## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены  
  
## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены

df <- data.frame(tmp)  
# add a prefix to the column names  
colnames(df) <- seq(1:8  
 )  
colnames(df) <- paste0("k",colnames(df))  
# get individual PCA  
df.pca <- prcomp(df, center = TRUE, scale. = FALSE)  
ind.coord <- df.pca$x  
ind.coord <- ind.coord[,1:2]  
df <- bind\_cols(as.data.frame(df), as.data.frame(ind.coord))  
clustree(df, prefix = "k")

## Warning: The `add` argument of `group\_by()` is deprecated as of dplyr 1.0.0.  
## Please use the `.add` argument instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

 ##Бачимо, що при розбитті на 2 кластери - маємо один кластер з більшою кількістю спостережень, інший - з меншою. важливо виділити те, що більшість спостережень, при кластеризації, потрапляють до одного кластеру. Більш рівномірна кількість спостережень у кластері буде 6 кластерах

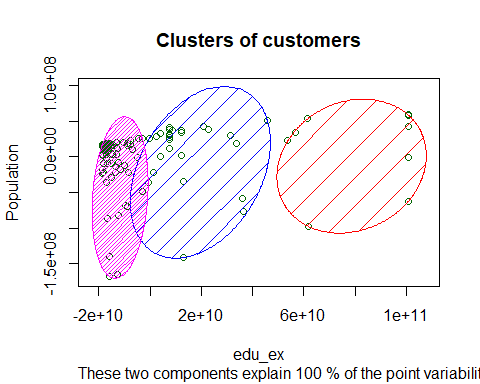
# Fitting K-Means to the dataset

set.seed(29)  
model\_km = kmeans(f, 3)  
#cluster cores  
y\_km = model\_km$cluster  
aggregate(f,by=list(y\_km),FUN=mean)

## Group.1 cr\_in Population gdp\_p unem life\_ex dea bir  
## 1 1 0.1818182 112892740 31984.18 6.234545 79.48764 8.503636 11.27664  
## 2 2 0.5517241 23822093 13377.79 7.211161 74.27737 7.530287 17.57883  
## 3 3 0.4400000 37698281 28606.99 8.078400 76.42088 8.100720 15.43860  
## emp emp\_s emp\_a emp\_i edu\_ex mil\_ex  
## 1 56.54455 67.73909 9.136364 23.12545 104296882869 39877921079  
## 2 57.43354 57.62311 20.640080 21.11115 4040650965 3619913684  
## 3 55.17332 66.80848 11.079880 21.97760 33578111642 10844871215

# Visualising the clusters

library(cluster)  
clusplot(f[,c('edu\_ex','Population')],  
 y\_km,  
 lines = 0,  
 shade = TRUE,  
 color = TRUE,  
 labels= 0,  
 plotchar = FALSE,  
 span = TRUE,  
 main = paste('Clusters of customers'),  
 xlab = 'edu\_ex',  
 ylab = 'Population')



# Comparing to HC

library(clusteval)

## Warning: package 'clusteval' was built under R version 3.6.3

cluster\_similarity(y\_hc,y\_km)

## [1] 0.924994

#Аналіз моделей показує, що спостереження стійкі до кластеризації, бо розбиття на кластери за першим та другим методами є подібними на 92.5%

mPopulation <- mean(f$Population)  
sPopulation <- sd(f$Population)  
f$Population <- (f$Population-mPopulation)/sPopulation  
  
mgdp\_p <- mean(f$gdp\_p)  
sgdp\_p <- sd(f$gdp\_p)  
f$gdp\_p<- (f$gdp\_p-mgdp\_p)/sgdp\_p  
  
munem <- mean(f$unem)  
sunem <- sd(f$unem)  
f$unem <- (f$unem-munem)/sunem  
  
mlife\_ex <- mean(f$life\_ex)  
slife\_ex <- sd(f$life\_ex)  
f$life\_ex <- (f$life\_ex-mlife\_ex)/slife\_ex  
  
mdea <- mean(f$dea)  
sdea <- sd(f$dea)  
f$dea<- (f$dea-mdea)/sdea  
  
mbir <- mean(f$bir)  
sbir <- sd(f$bir)  
f$bir <- (f$bir-mbir)/sbir  
  
memp <- mean(f$emp)  
semp <- sd(f$emp)  
f$emp <- (f$emp-memp)/semp  
  
memp\_s <- mean(f$emp\_s)  
semp\_s <- sd(f$emp\_s)  
f$emp\_s<- (f$emp\_s-memp\_s)/semp\_s  
  
memp\_a <- mean(f$emp\_a)  
semp\_a <- sd(f$emp\_a)  
f$emp\_a <- (f$emp\_a-memp\_a)/semp\_a  
  
memp\_i <- mean(f$emp\_i)  
semp\_i <- sd(f$emp\_i)  
f$emp\_i <- (f$emp\_i-memp\_i)/semp\_i  
  
medu\_ex <- mean(f$edu\_ex)  
sedu\_ex <- sd(f$edu\_ex)  
f$edu\_ex<- (f$edu\_ex-medu\_ex)/sedu\_ex  
  
mmil\_ex <- mean(f$mil\_ex)  
smil\_ex <- sd(f$mil\_ex)  
f$mil\_ex <- (f$mil\_ex-mmil\_ex)/smil\_ex  
  
f\_matrix <- as.matrix(f)  
head (f)

## cr\_in Population gdp\_p unem life\_ex dea bir  
## 1 1 0.03404554 -0.9191786 0.8472187 -1.92163256 -0.428923107 2.23006043  
## 2 0 -0.63995695 -0.7469257 1.4084049 0.54801921 -0.008656897 -0.62521158  
## 3 1 0.13674621 -0.7736459 1.0262742 0.22974718 -1.120688727 1.11185648  
## 4 1 -0.09662224 -0.7708171 -0.0393244 -2.10703037 0.259123343 2.23006043  
## 5 1 0.19029893 -0.1946230 0.2292589 0.20713769 -0.042129427 0.08467382  
## 6 0 -0.63851808 -0.7807704 2.2665612 -0.06591535 0.790964652 -0.30656984  
## emp emp\_s emp\_a emp\_i edu\_ex mil\_ex  
## 1 -1.3054994 -1.4645832 1.5143179 -0.5351602 -0.6024304 -0.6393203  
## 2 -0.8281562 -1.1736896 1.1812142 -0.3296526 -0.6056667 -0.6430138  
## 3 -1.6591152 -0.1009786 -0.4319510 1.4825507 -0.3797282 0.1368544  
## 4 1.4468612 -1.2632456 1.8922956 -2.0162161 -0.4863262 -0.4137821  
## 5 -0.1302805 1.1181597 -0.9523177 0.1514220 0.3998826 -0.2255286  
## 6 -1.0591903 -0.5520271 0.7865525 -0.7406678 -0.6071068 -0.6195111

#Висновок: шкалювання кількісних змінних

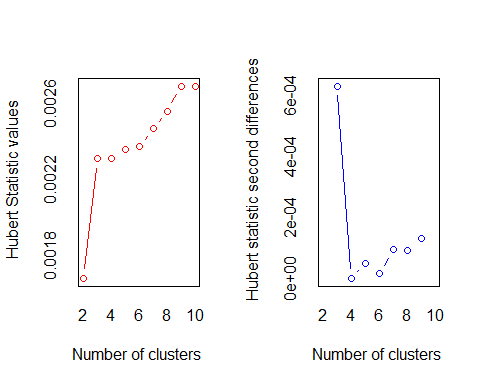
# NbCLust

library(factoextra)

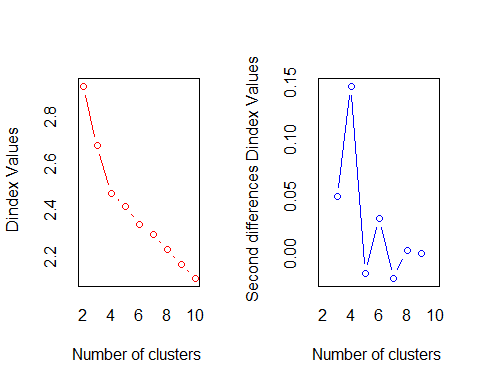
## Warning: package 'factoextra' was built under R version 3.6.3

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

library(NbClust)  
res.nbclust <- NbClust(f, distance = "euclidean",  
 min.nc = 2, max.nc = 10,   
 method = "complete", index ="all")



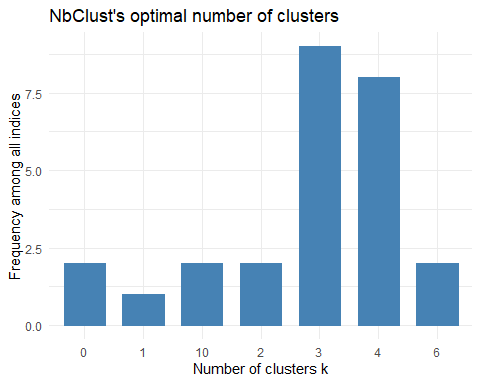
## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.  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



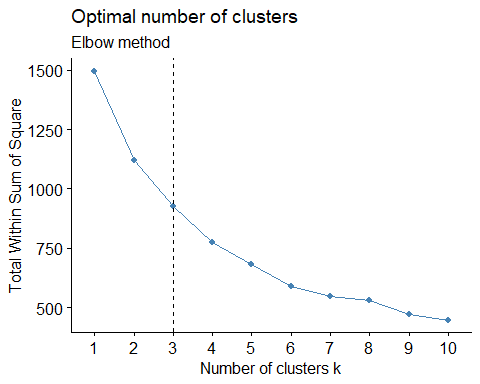
## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 2 proposed 2 as the best number of clusters   
## \* 9 proposed 3 as the best number of clusters   
## \* 8 proposed 4 as the best number of clusters   
## \* 2 proposed 6 as the best number of clusters   
## \* 2 proposed 10 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 3   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fviz\_nbclust(res.nbclust) + theme\_minimal() + ggtitle("NbClust's optimal number of clusters")

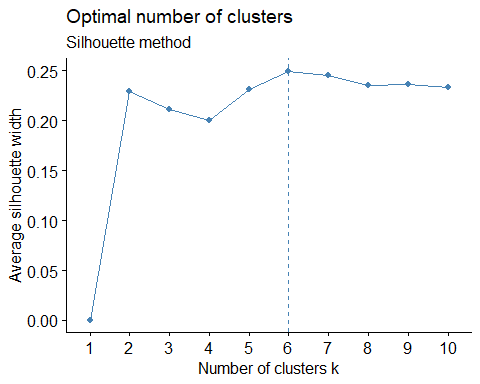
## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 1 proposed 1 as the best number of clusters  
## \* 2 proposed 2 as the best number of clusters  
## \* 9 proposed 3 as the best number of clusters  
## \* 8 proposed 4 as the best number of clusters  
## \* 2 proposed 6 as the best number of clusters  
## \* 2 proposed 10 as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 3 .



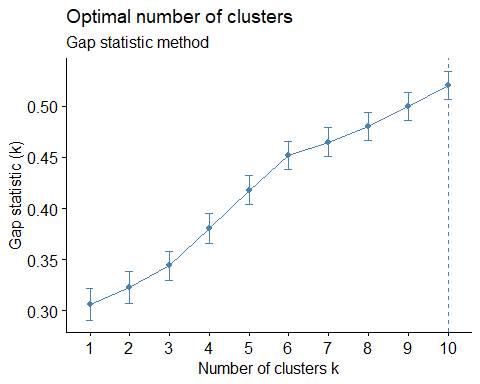
# Elbow method  
# The sum of squares at each number of clusters is calculated and graphed, and the user looks for a change of slope from steep to shallow (an elbow) to determine the optimal number of clusters.  
fviz\_nbclust(f, kmeans, method = "wss") +  
 geom\_vline(xintercept = 3, linetype = 2)+  
 labs(subtitle = "Elbow method")



# Silhouette method  
# The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for k.  
fviz\_nbclust(f, kmeans, method = "silhouette")+  
 labs(subtitle = "Silhouette method")



# Gap statistic  
# The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The estimate of the optimal clusters will be value that maximize the gap statistic. This means that the clustering structure is far away from the random uniform distribution of points.  
fviz\_nbclust(f, kmeans, nstart = 25, method = "gap\_stat", nboot = 50)+  
 labs(subtitle = "Gap statistic method")

 ##бачимо з аналізу, що 9 голосів надано за 3 кластери, 8 голосів за 4 кластери. Зупинимось на 3 кластерах.

#“Kohonen maps” # Features Scaling

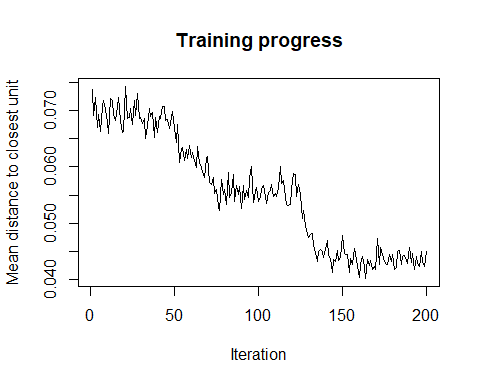
#Висновок: моделі класифікації вимагають попереднього шкалювання кількісних змінних. Шкалювання виконано.

# Fitting the NN

set.seed(123)  
library(kohonen)

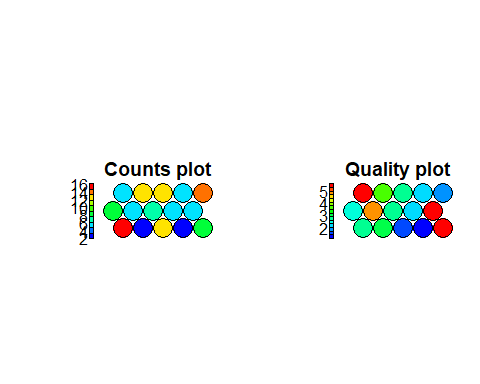
## Warning: package 'kohonen' was built under R version 3.6.3

som\_grid <- somgrid(xdim = 5, ydim = 3, topo = "hexagonal")   
som\_model <- som(f\_matrix, grid = som\_grid, rlen = 200,  
 alpha = c(0.05,0.01), keep.data = TRUE)  
plot(som\_model, type = "changes")

 #Висновок: модель Кохонена потребуює шкалювання, шкалювання кількісних змінних. Шкалювання виконано. #Висновок: графік зміни помилки нейронної мережі Кохонена свідчить про успішне навчання моделі.

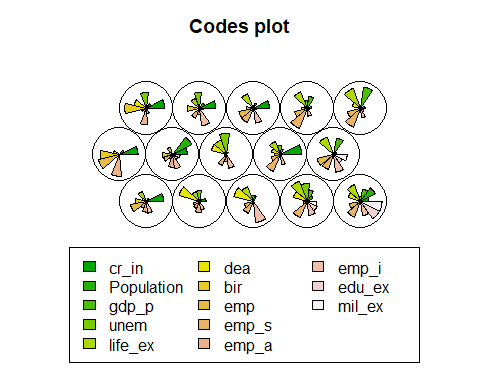
## Visualization

#Palette  
coolBlueHotRed <- function(n, alpha = 1) {  
 rainbow(n, end = 4/6, alpha = alpha)[n:1]   
}  
par(mfrow = c(1, 2))  
#Number of objects at sells  
plot(som\_model, type = "counts", palette.name = coolBlueHotRed)  
#Distance to core  
plot(som\_model, type = "quality", palette.name = coolBlueHotRed)

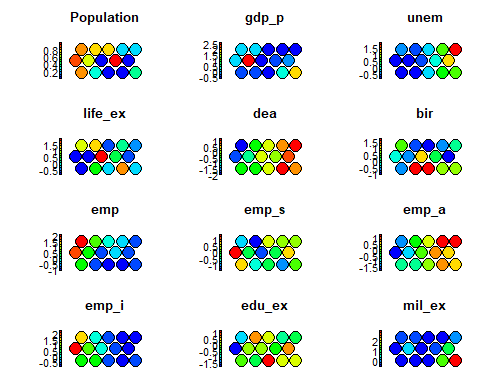
 #Висновок: в кожну клітинку мережі потрапило від 2 до 16 спостережень, мережа досить повна.

## Maps of the factors

plot(som\_model, type = "codes")

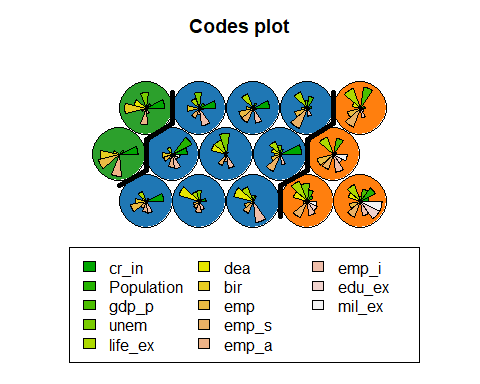
 #Висновок: ця візуалізація дозволяє аналізувати всі фактори на одній карті.

par(mfrow = c(4, 3))  
  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,1],  
 main = "Population",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,2],   
 main = "gdp\_p",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,3],   
 main = "unem",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,4],   
 main = "life\_ex",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,5],   
 main = "dea",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,6],   
 main = "bir",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,7],   
 main = "emp",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,8],   
 main = "emp\_s",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,9],   
 main = "emp\_a",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,10],   
 main = "emp\_i",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,11],   
 main = "edu\_ex",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,12],   
 main = "mil\_ex",  
 palette.name = coolBlueHotRed)

 #Висновок: ця візуалізація дозволяє аналізувати всі фактори на окремих картах.

# Clusters description

mydata <- as.matrix(som\_model$codes[[1]])  
#Use hierarchical clustering, k=3  
som\_cluster <- cutree(hclust(dist(mydata)), 3)  
#Palette  
pretty\_palette <- c("#1f77b4", '#ff7f0e', '#2ca02c',  
 '#d62728', '#9467bd', '#8c564b', '#e377c2')  
#Colored clusters  
plot(som\_model, type = "codes",   
 bgcol = pretty\_palette[som\_cluster])  
add.cluster.boundaries(som\_model, som\_cluster)

 #Висновок: ця візуалізація дозволяє аналізувати парамерти двох виявлених кластерів на одній карті. #```{r}

fPopulation\*sPopulation fPopulation+mPopulation

fgdp\_p\*sgdp\_p fgdp\_p+mgdp\_p

funem\*sunem funem+munem

flife\_ex\*slife\_ex flife\_ex+mlife\_ex

fdea\*sdea fdea+mdea

fbir\*sbir fbir+mbir

femp\*semp femp+memp

femp\_s\*semp\_s femp\_s+memp\_s

femp\_a\*semp\_a femp\_a+memp\_a

femp\_i\*semp\_i femp\_i+memp\_i

fedu\_ex\*sedu\_ex fedu\_ex+medu\_ex

fmil\_ex\*smil\_ex fmil\_ex+mmil\_ex

head (f) #```

aggregate(mydata,by=list(som\_cluster),FUN=mean)

## Group.1 cr\_in Population gdp\_p unem life\_ex dea  
## 1 1 0.4974845 -0.04579547 -0.3728181 0.23549015 -0.04159624 0.1142789  
## 2 2 0.1506893 0.37304594 1.1500706 0.02668789 0.97792435 0.1273089  
## 3 3 0.8645333 0.15540493 -0.6905326 0.17643415 -1.90548203 0.1750959  
## bir emp emp\_s emp\_a emp\_i edu\_ex  
## 1 -0.1291324 -0.26305128 -0.01812121 0.004692508 0.08019583 -0.3222468  
## 2 -0.8242662 0.07167125 0.78582433 -0.795060496 0.17745250 1.3610312  
## 3 1.9820402 0.28791682 -1.58357293 1.937498357 -1.48280588 -0.3995056  
## mil\_ex  
## 1 -0.2485137  
## 2 1.2609267  
## 3 -0.5811879

#Висновок: на основі нейроних мереж Кохонена виявлено три кластери. Розраховано характеристики типових об’єктів кластерів. #Отже, кластерізацію проведено за трьома методами , а саме: Ієрархічна кластеризація, Кластеризація на основі k-means та на основі нейроних мереж Кохонена. Аналіз моделей показує, що спостереження стійкі до кластеризації, бо розбиття на кластери за першим та другим методами є подібними на 92.4%. За 3-ма методами було виділено 3 кластери.