HOMEWORK 7: Modern Statistics and Big Data Analytics

Federico Veronesi

2023-11-26

Exercise 1

Data prep

```
library(poLCA)

data(election)
election12 <- election[,1:12]

### create a new factor level for missing values
electionwithna <- election12
for (i in 1:12){
levels(electionwithna[,i]) <- c(levels(election12[,i]),"NA")
electionwithna[is.na(election12[,i]),i] <- "NA"
}
```

Point a: latent class model with poLCA (3 classes)

```
### MDS election data
library(smacof)

library(cluster)
SM_dist <- daisy(electionwithna, metric = "gower")
mds_election <- mds(SM_dist, ndim = 2)

f <-
cbind(MORALG,CARESG,KNOWG,LEADG,DISHONG,INTELG,MORALB,CARESB,KNOWB,LEADB,DISHONB,I
NTELB)~1
LCM1 <- poLCA (f, electionwithna, nclass=3, maxiter=1000, na.rm = F, nrep = 10)

## Model 1: llik = -25887.51 ... best llik = -25887.51
## Model 2: llik = -25885.25 ... best llik = -25885.25
## Model 3: llik = -25887.51 ... best llik = -25885.25
## Model 4: llik = -25885.25 ... best llik = -25885.25
## Model 5: llik = -25990.17 ... best llik = -25885.25
## Model 6: llik = -25891.5 ... best llik = -25885.25
## Model 7: llik = -25891.5 ... best llik = -25885.25
## Model 8: llik = -25990.17 ... best llik = -25885.25
## Model 9: llik = -25972.59 ... best llik = -25885.25
## Model 10: llik = -25891.91 ... best llik = -25885.25
## Conditional item response (column) probabilities,
##  by outcome variable, for each class (row)
##
## $MORALG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.1069       0.4510         0.2755            0.1321 0.0345
## class 2:           0.0576       0.4082         0.1332            0.0660 0.3351
## class 3:           0.4680       0.4895         0.0286            0.0060 0.0079
```

```
##
## $CARESG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.0333       0.3201         0.4160            0.2051 0.0255
## class 2:           0.0769       0.2832         0.2050            0.1594 0.2756
## class 3:           0.3371       0.5429         0.0867            0.0252 0.0081
##
## $KNOWG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.1194       0.6230         0.1978            0.0496 0.0102
## class 2:           0.0818       0.5479         0.1376            0.0513 0.1814
## class 3:           0.4990       0.4824         0.0131            0.0055 0.0000
##
## $LEADG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.0282       0.3112         0.4574            0.1851 0.0179
## class 2:           0.0394       0.3353         0.2541            0.0966 0.2746
## class 3:           0.3298       0.5560         0.1017            0.0049 0.0076
##
## $DISHONG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.1248       0.2687         0.3919            0.1771 0.0375
## class 2:           0.0412       0.1604         0.2142            0.1767 0.4075
## class 3:           0.0247       0.0634         0.3563            0.5322 0.0235
##
## $INTELG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.1511       0.6238         0.1649            0.0509 0.0093
## class 2:           0.0876       0.5720         0.1144            0.0683 0.1577
## class 3:           0.5060       0.4691         0.0188            0.0061 0.0000
##
## $MORALB
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.3151       0.5678         0.0997            0.0042 0.0132
## class 2:           0.0389       0.2634         0.1132            0.1183 0.4661
## class 3:           0.0936       0.4307         0.3185            0.0937 0.0635
##
## $CARESB
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.1625       0.5842         0.2230            0.0213 0.0090
## class 2:           0.0253       0.1451         0.2268            0.2826 0.3203
## class 3:           0.0162       0.1375         0.4631            0.3687 0.0144
##
## $KNOWB
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.2390       0.6724         0.0856            0.0009 0.0021
## class 2:           0.0836       0.4037         0.1669            0.1275 0.2183
## class 3:           0.0740       0.3821         0.3875            0.1457 0.0107
##
## $LEADB
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
```

```
## class 1:            0.2759       0.6415         0.0703            0.0041 0.0082
## class 2:            0.0347       0.3396         0.1811            0.1566 0.2881
## class 3:            0.0350       0.3111         0.4425            0.1792 0.0323
##
## $DISHONB
##           1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:            0.0180       0.0933         0.3758            0.4861 0.0269
## class 2:            0.0478       0.1546         0.2031            0.1219 0.4725
## class 3:            0.0623       0.2487         0.4161            0.1935 0.0795
##
## $INTELB
##           1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:            0.2791       0.6565         0.0637            0.0007 0.0000
## class 2:            0.0591       0.4647         0.1266            0.1001 0.2495
## class 3:            0.1144       0.4285         0.3229            0.1227 0.0116
##
## Estimated class population shares
##  0.4735 0.1462 0.3803
##
## Predicted class memberships (by modal posterior prob.)
##  0.4756 0.1429 0.3815
##
## =========================================================
## Fit for 3 latent classes:
## =========================================================
## number of observations: 1785
## number of estimated parameters: 146
## residual degrees of freedom: 1639
## maximum log-likelihood: -25885.25
##
## AIC(3): 52062.51
## BIC(3): 52863.64
## G^2(3): 25461.92 (Likelihood ratio/deviance statistic)
## X^2(3): 12552843831 (Chi-square goodness of fit)
##

plot(mds_election$conf, col=LCM1$predclass, main = paste("MDS plot: latent class
(stress=", round(mds_election$stress,3)*100, "%)"))
```

Results: the interpretation of the "red" cluster is difficult (not homogeneous). The other two clusters are homegeneous and not well separated. The MDS loses a lot of information (31%) so the plot could somehow be misleading.

Point b: latent class model with flexmixedruns (3 classes)

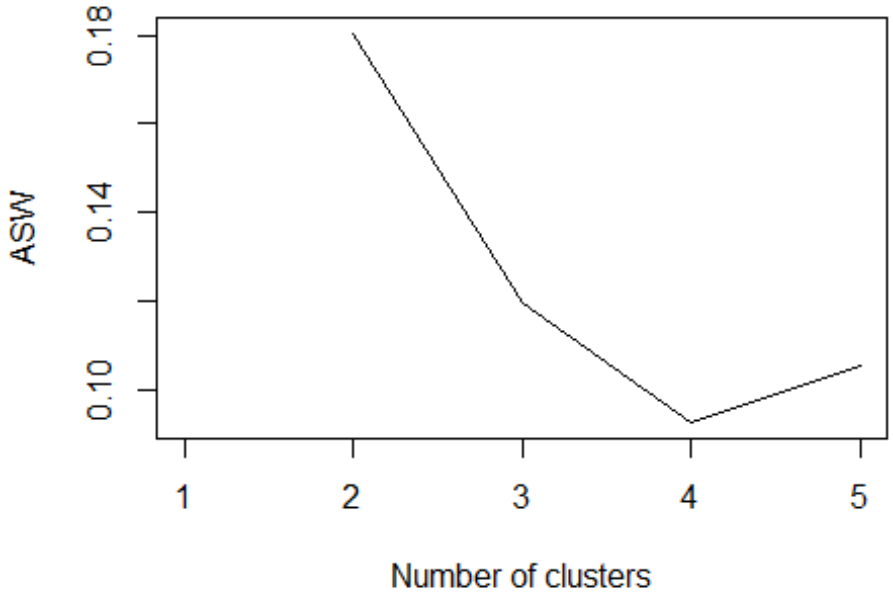
```
library(flexmix)

## Warning: il pacchetto 'flexmix' è stato creato con R versione 4.2.3

## Caricamento del pacchetto richiesto: lattice

library(fpc)

## Warning: il pacchetto 'fpc' è stato creato con R versione 4.2.3

LCM2 <- flexmixedruns(electionwithna, continuous = 0, discrete =12, n.cluster = 3)

## k=  3  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## k=  3  new best fit found in run  3
## k=  3  new best fit found in run  4
## Nonoptimal or repeated fit found in run  5
## Nonoptimal or repeated fit found in run  6
## Nonoptimal or repeated fit found in run  7
## Nonoptimal or repeated fit found in run  8
## Nonoptimal or repeated fit found in run  9
## Nonoptimal or repeated fit found in run  10
## Nonoptimal or repeated fit found in run  11
## Nonoptimal or repeated fit found in run  12
```

```
## k=  3  new best fit found in run  13
## Nonoptimal or repeated fit found in run  14
## Nonoptimal or repeated fit found in run  15
## Nonoptimal or repeated fit found in run  16
## Nonoptimal or repeated fit found in run  17
## Nonoptimal or repeated fit found in run  18
## Nonoptimal or repeated fit found in run  19
## Nonoptimal or repeated fit found in run  20
## k=  3  BIC=  52863.68

plot(mds_election$conf, col=LCM2$flexout[[3]]@cluster ,main = paste("MDS plot:
latent class (stress=", round(mds_election$stress,3)*100, "%)"))
```

The results are very similar to the previous point. So we can say that in this case there's no big difference between the two functions.

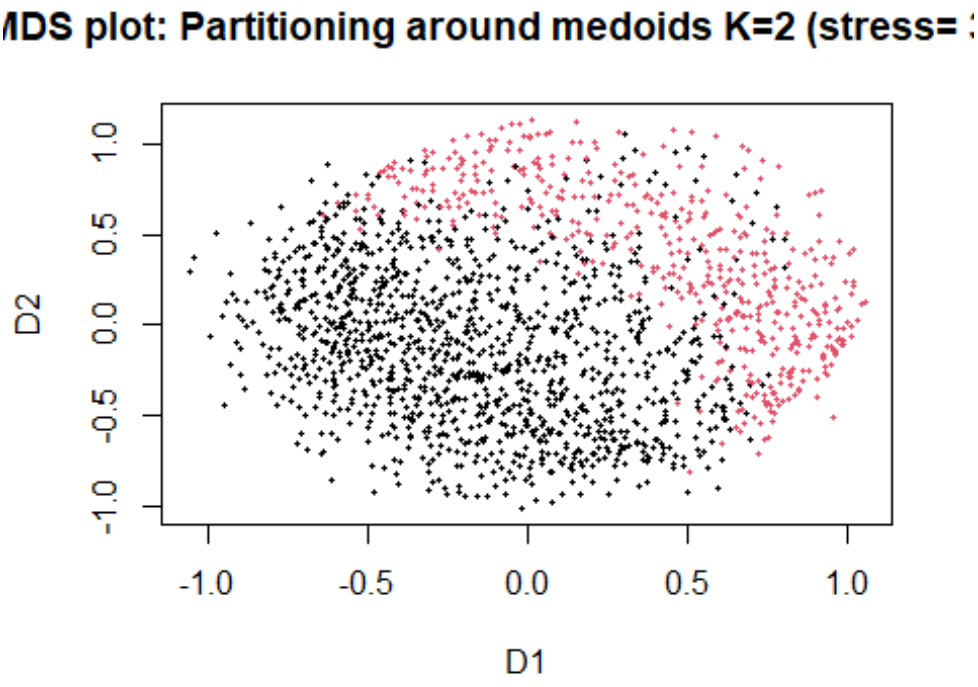
Point c: Partitioning Around Medoids

```
library(cluster)
election_pam <- list()
sil <- list()
asw <- c()
for (k in 2:5) {election_pam[[k]] <- pam(SM_dist, k)
                sil[[k]] <- silhouette(election_pam[[k]],dist=SM_dist)
                asw[k] <- summary(sil[[k]])$avg.width
                }
plot(1:5,asw,type="l",xlab="Number of clusters",ylab="ASW")
```

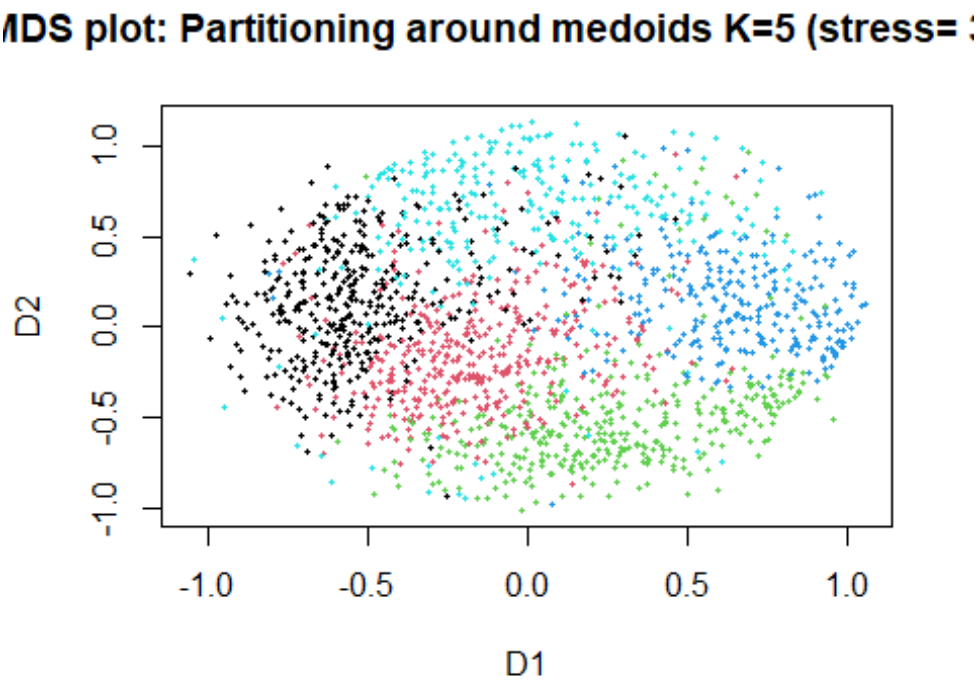


K=2 seems the best n°of clusters, but we will investigate also K=5 as a local optimum.

```
plot(mds_election$conf, col=election_pam[[2]]$clustering ,main = paste("MDS plot:
Partitioning around medoids K=2 (stress=", round(mds_election$stress,3)*100,
"%)"), pch=20,cex=0.7)
```



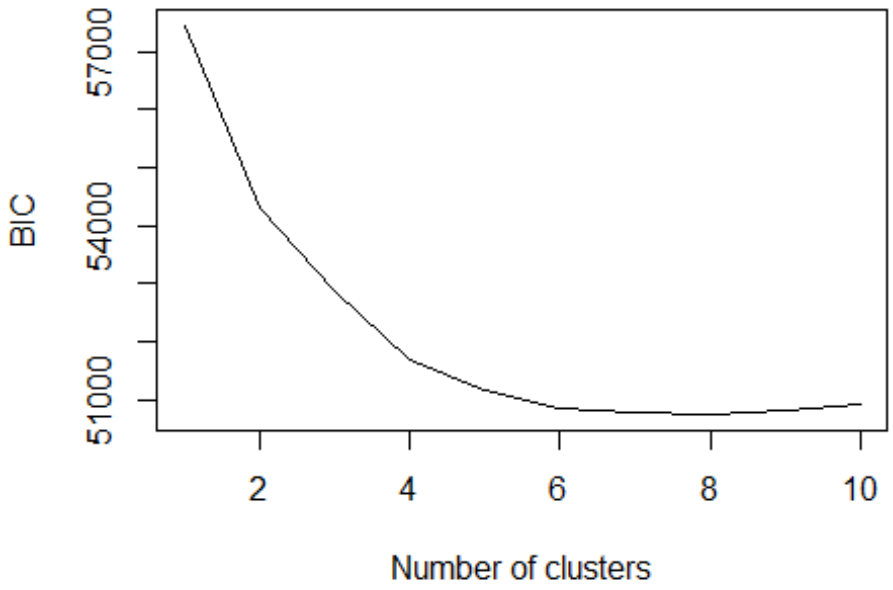
```
plot(mds_election$conf, col=election_pam[[5]]$clustering ,main = paste("MDS plot:
Partitioning around medoids K=5 (stress=", round(mds_election$stress,3)*100,
"%)"), pch=20,cex=0.7)
```



The solution with 5 clusters is quite good (it produces homogeneous clusters). However I would choose K=2. It depends on our will to have smaller clusters (in this case, K=5), or two big clusters (K=2)

Point d

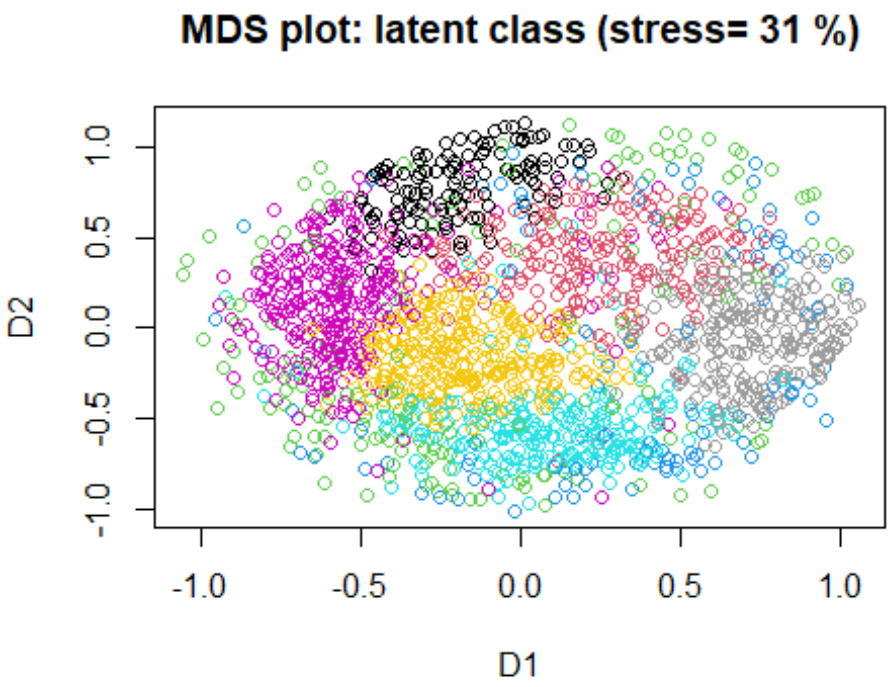
```
LCMflex <- flexmixedruns(electionwithna, continuous = 0, discrete =12, n.cluster =
1:10)

plot(1:10,LCMflex$bicvals,typ="l", xlab="Number of clusters",ylab="BIC")
```




```
which.min(LCMflex$bicvals)

## [1] 8
```

We have the smallest BIC for k=8:

```
plot(mds_election$conf, col=LCMflex$flexout[[8]]@cluster ,main = paste("MDS plot:
latent class (stress=", round(mds_election$stress,3)*100, "%)"))
```



There is overlapping between the components in certain zones of the plot. Maybe, a smaller K could be investigated, even if BIC suggests K=8

Point e:

CODE: election[is.na(electionAGE),14] $< -mean(election$AGE)
election[is.na(election$EDUC$),15] $< -mean(election$EDUC)

for (i in 1:12){ levels(election[,i]) <- c(levels(election[,i]),"NA") election[is.na(election[,i]),i] <- "NA" }

election <- election [,c(14:15), c(1:12)]
LCMflex <- flexmixedruns(election, continuous = 2, discrete =12, n.cluster = 1:10)

The function flexmixedruns gives an error when we insert some continuous variable. I wasn't able to solve it.

Exercise 2

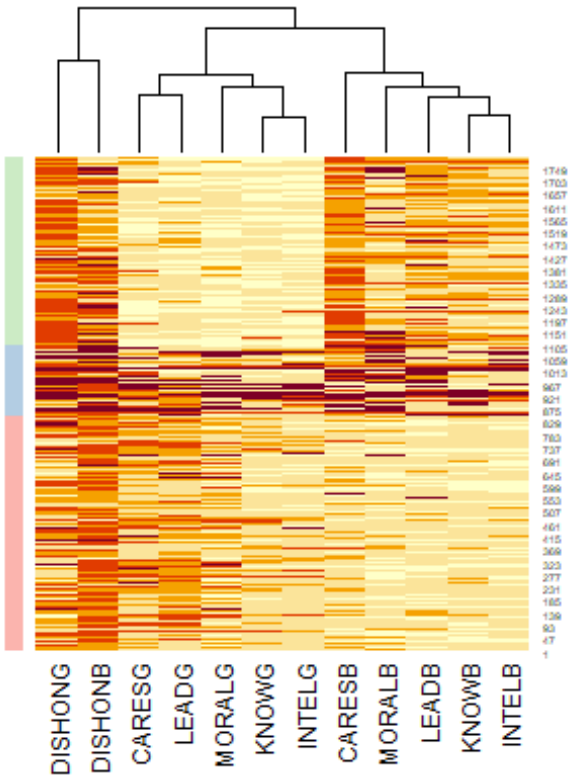
Heatmap

```
library(RColorBrewer)
mat_e12na = as.matrix(electionwithna)
mat_e12na <- ifelse(mat_e12na == "1 Extremely well", 1,
                            ifelse(mat_e12na == "2 Quite well", 2,
                                   ifelse(mat_e12na == "3 Not too well", 3,
                                          ifelse(mat_e12na == "4 Not well at all",
4, 5))))
mat_e12na = cbind(mat_e12na, LCM2$flexout[[3]]@cluster)
```

```r
colnames(mat_e12na)[13] = "CLASS"
mat_e12na_O = mat_e12na[order(mat_e12na[,13]),]
for (col in 1:12){
  mat_e12na_O[, col] <- as.factor(mat_e12na_O[,col])
}

mat_dist <- daisy(t(mat_e12na_O[,-13]),metric="gower")
```

```
## Warning in daisy(t(mat_e12na_O[, -13]), metric = "gower"): variabili binarie
## 5, 11, 14, 17, 23, 27, 33, 47, 52, 55, 58, 63, 65, 73, 81, 87, 89, 91, 97,
## 106, 112, 118, 133, 134, 144, 154, 166, 168, 170, 182, 189, 192, 195, 207, 213,
## 216, 217, 219, 240, 242, 246, 253, 265, 266, 273, 281, 282, 294, 303, 305, 309,
## 311, 315, 321, 328, 330, 334, 343, 347, 348, 349, 350, 352, 364, 367, 368, 373,
## 382, 385, 389, 395, 402, 406, 407, 422, 433, 434, 435, 438, 450, 457, 470, 491,
## 504, 508, 519, 522, 524, 525, 529, 530, 532, 541, 547, 556, 557, 561, 563, 568,
## 571, 573, 580, 584, 589, 594, 616, 620, 623, 624, 628, 630, 631, 637, 645, 649,
## 659, 660, 664, 667, 669, 673, 710, 720, 728, 731, 735, 739, 756, 764, 773, 776,
## 782, 790, 794, 796, 804, 812, 817, 841, 846, 865, 866, 868, 869, 874, 877, 881,
## 896, 903, 907, 916, 920, 928, 936, 958, 960, 963, 965, 966, 968, 971, 972, 977,
## 983, 994, 1000, 1015, 1017, 1023, 1030, 1036, 1040, 1061, 1069, 1085, 1094,
## 1098, 1109, 1121, 1208, 1214, 1228, 1229, 1268, 1280, 1286, 1291, 1298, 1299,
## 1306, 1311, 1314, 1320, 1324, 1325, 1327, 1341, 1343, 1355, 1359, 1363, 1365,
## 1379, 1410, 1414, 1415, 1416, 1428, 1457, 1458, 1460, 1495, 1496, 1508, 1510,
## 1525, 1530, 1585, 1602, 1608, 1615, 1619, 1628, 1631, 1635, 1637, 1653, 1655,
## 1665, 1677, 1699, 1705, 1726, 1736, 1738, 1743, 1753, 1754, 1761 trattate come
## intervallo ridimensionato
```

```r
varclust <- hclust(mat_dist,method="complete")


col2 <- brewer.pal(5, "Pastel1")
heatmap(mat_e12na_O[,-13], Rowv=NA, RowSideColors=col2[mat_e12na_O[,13]] ,
Colv=as.dendrogram(varclust),scale="none")
```



From this heatmap, the conditional independence seems to hold: there are three quite well dinstinct patterns corresponding to the clusters.

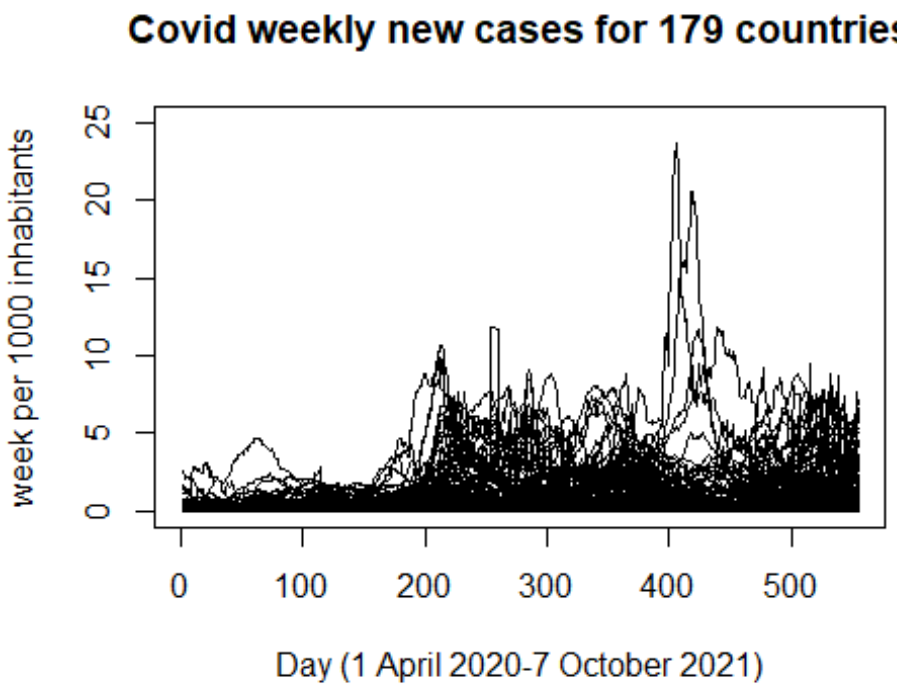
Exercise 3

a) number of free parameters: $(2-1)^5 * (3-1)^3 * (5-1)^2 = 128$

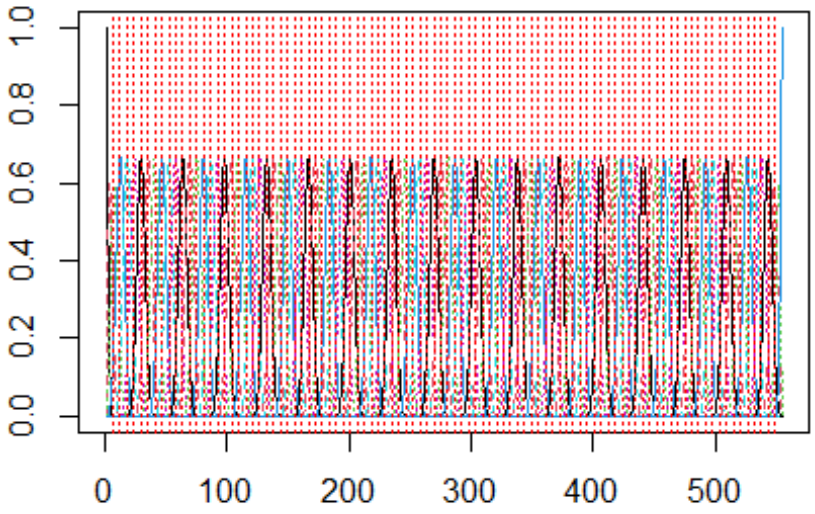
b) n°of free parameters = $(K-1) + K\left(\sum_{j=1}^{p} m_j - 1\right) = 3 + 4(1+1+1+1+1+2+2+2+4+4) = 39$

Exercise 4a

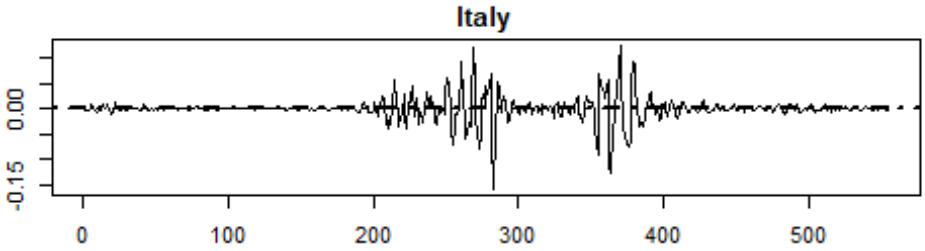
```
setwd("C:/Users/Veronesi/Desktop/uniBo/Magistrale/Modern Statistics and Big Data
Analytics")

library(fda) # Functional data analysis

covid21 <- read.table("covid2021.dat")
covid <- read.table("covid2021.dat")
covid21v <- as.matrix(covid21[,5:559])
# Raw data plot:
plot(1:555,covid21v[1,],type="l",ylim=c(0,25),ylab="New cases over one
week per 1000 inhabitants",xlab="Day (1 April 2020-7 October 2021)",
main="Covid weekly new cases for 179 countries")
for(i in 2:179)
points(1:555,covid21v[i,],type="l")
```



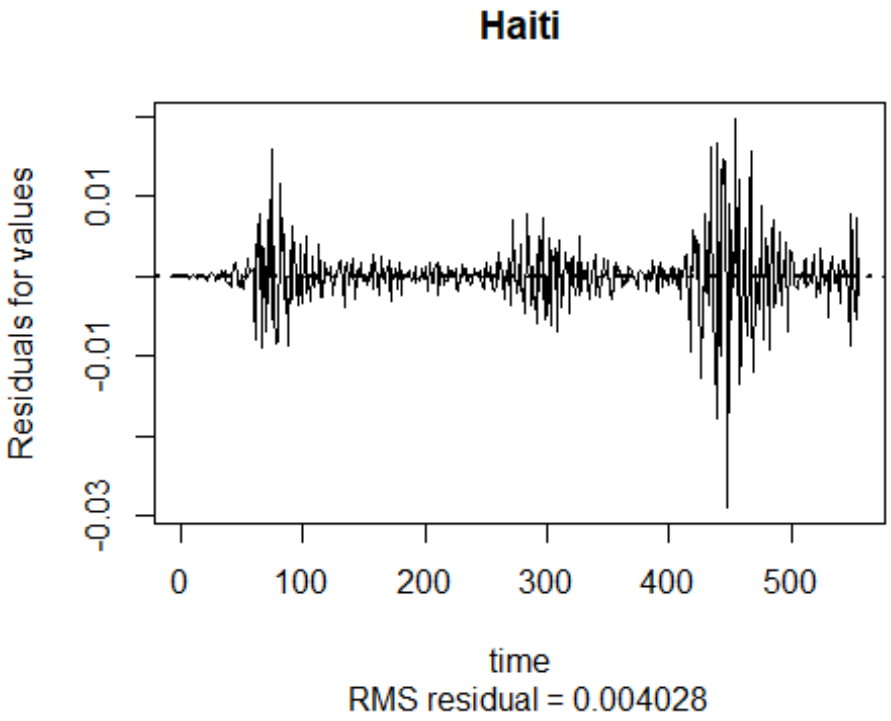

```
# Constructing B-spline basis
bbasis100 <- create.bspline.basis(c(1,555),nbasis=100) #with p=100
fdcovid100 <- Data2fd(1:555,y=t(as.matrix(covid21v)),basisobj=bbasis100)
# Plot basis
plot(bbasis100)
```



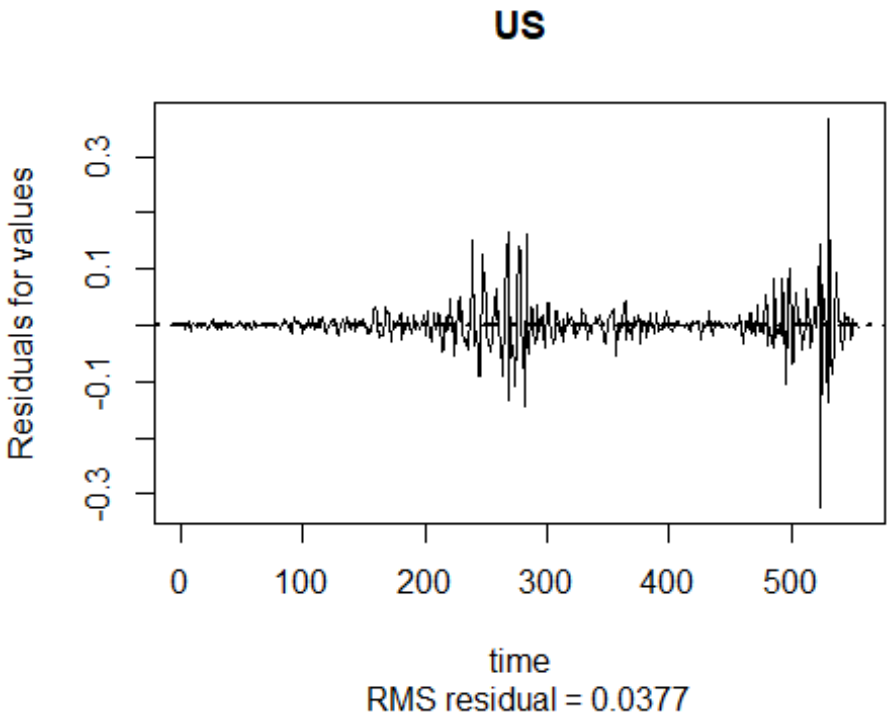

```
par(mfrow = c(3,1), mar = c(2,2,2,2))
plotfit.fd(t(covid21v),1:555,fdcovid100,index=79,cex.pch=0.5, residual = TRUE)
```




```
plotfit.fd(t(covid21v),1:555,fdcovid100,index=69,cex.pch=0.5,residual= TRUE)
```



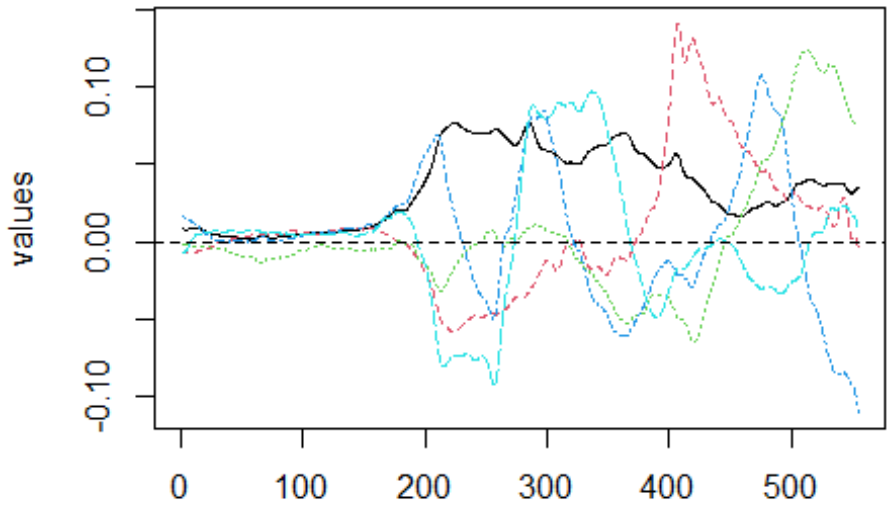

```
plotfit.fd(t(covid21v),1:555,fdcovid100,index=164,cex.pch=0.5,residual= TRUE)
```



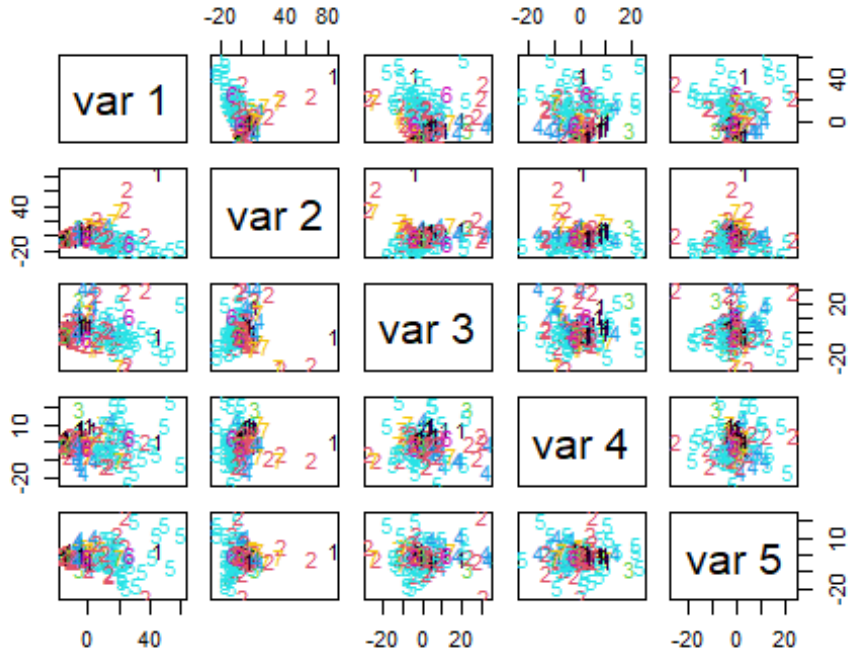
The residuals are scattered around zero, but their variance isn't constant over time. This means that there are potential violations of model assumptions.

Exercise 4b

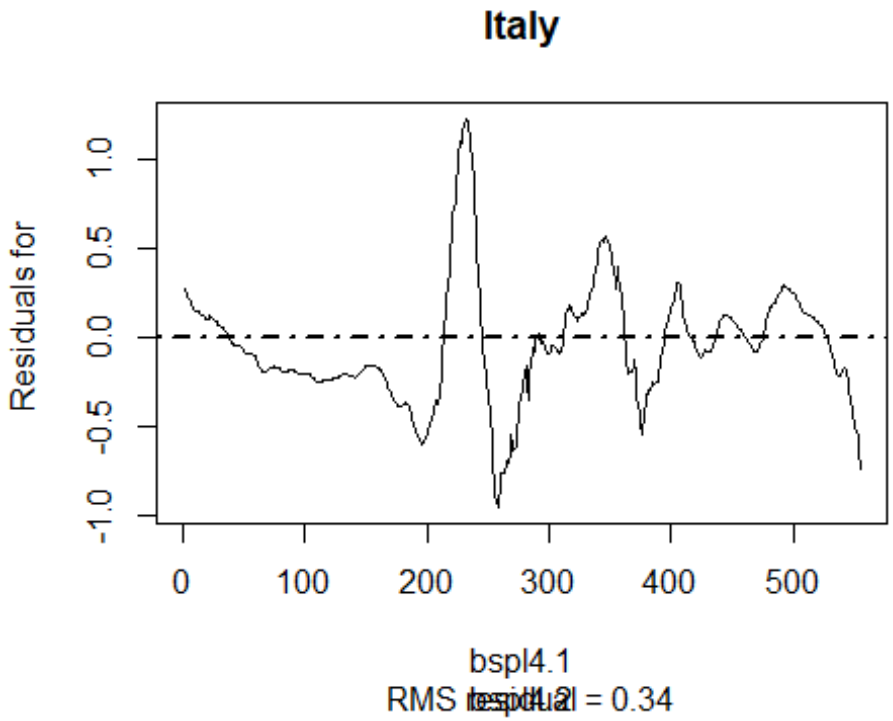
```
fdcovid <- Data2fd(1:555,y=t(as.matrix(covid21v)),basisobj=bbasis100)
covidpca <- pca.fd(fdcovid, nharm = 5)
plot(covidpca$harmonics) # PCs phi_k
```



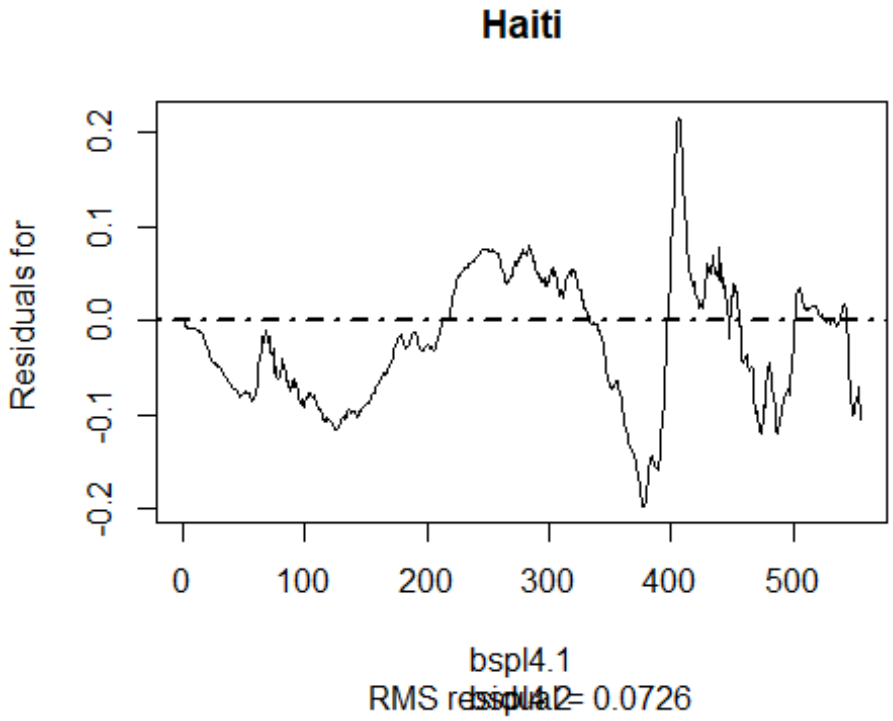

```
## [1] "done"

ncontinent <- as.numeric(as.factor(covid21$continent))
levels(as.factor(covid21$continent))

## [1] "Africa"          "Asia"            "Australia"       "Central America"
## [5] "Europe"          "North America"   "South America"

#[1] "Africa" "Asia" "Australia"
# "Central America"
#[5] "Europe" "North America" "South America"
pairs(covidpca$scores,col=ncontinent,pch=clusym[ncontinent])
```



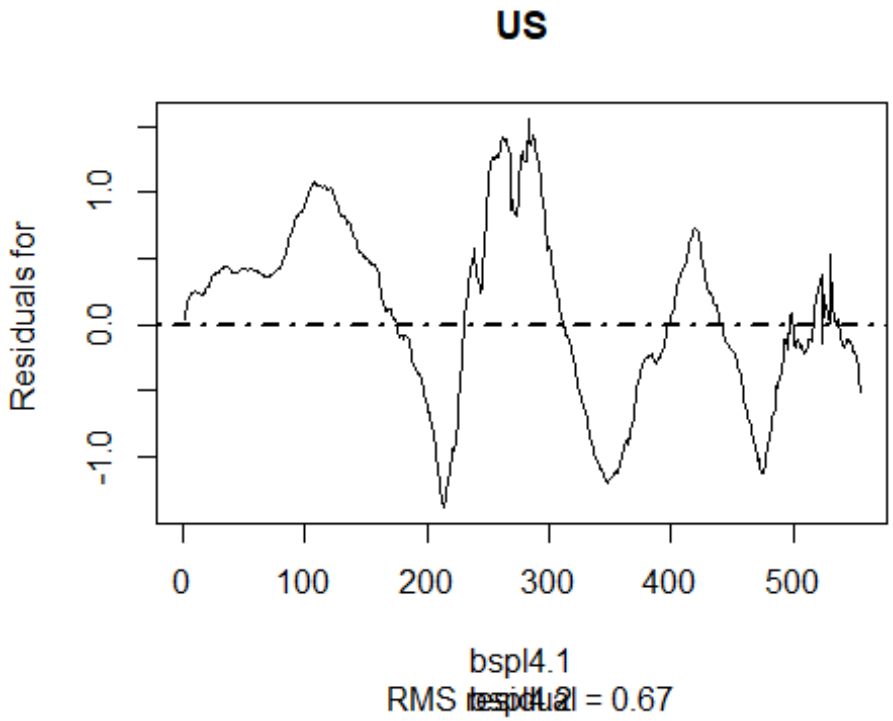

```r
# PCA scores

# Create functional data object of PCA approximations
mcovid <- mean.fd(fdcovid)

covidpcaapprox <- covidpca$harmonics
i <- 1
pcacoefi <- covidpca$harmonics$coefs %*% covidpca$scores[i,]+mcovid$coefs
covidpcaapprox$coefs <- pcacoefi
for (i in 2:179){
pcacoefi <- covidpca$harmonics$coefs %*% covidpca$scores[i,]+mcovid$coefs
covidpcaapprox$coefs <- cbind(covidpcaapprox$coefs, pcacoefi)
}
dimnames(covidpcaapprox$coefs)[[2]] <- covid21[,1]
plotfit.fd(t(covid21v),1:555,covidpcaapprox,index=79,cex.pch=0.5, residual = T)
```




```
plotfit.fd(t(covid21v),1:555,covidpcaapprox,index=69,cex.pch=0.5, residual = T)
```




```
plotfit.fd(t(covid21v),1:555,covidpcaapprox,index=164,cex.pch=0.5, residual = T)
```



Here the residuals are scattered around 0 and the magnitude of their variance seems constant over time. We can say that model assumptions are not violated.

Exercise 5

```
covidpca1 <- pca.fd(fdcovid, nharm = 1)
anova_data <- cbind(covidpca1$scores, covid$continent)
anova_data <- as.data.frame(anova_data)
anova_data[,1] <- as.numeric(anova_data[,1])
colnames(anova_data) <- c("firstPC", "continent")
onewayanova <- aov(formula = firstPC ~ continent, data = anova_data)
summary(onewayanova)
```

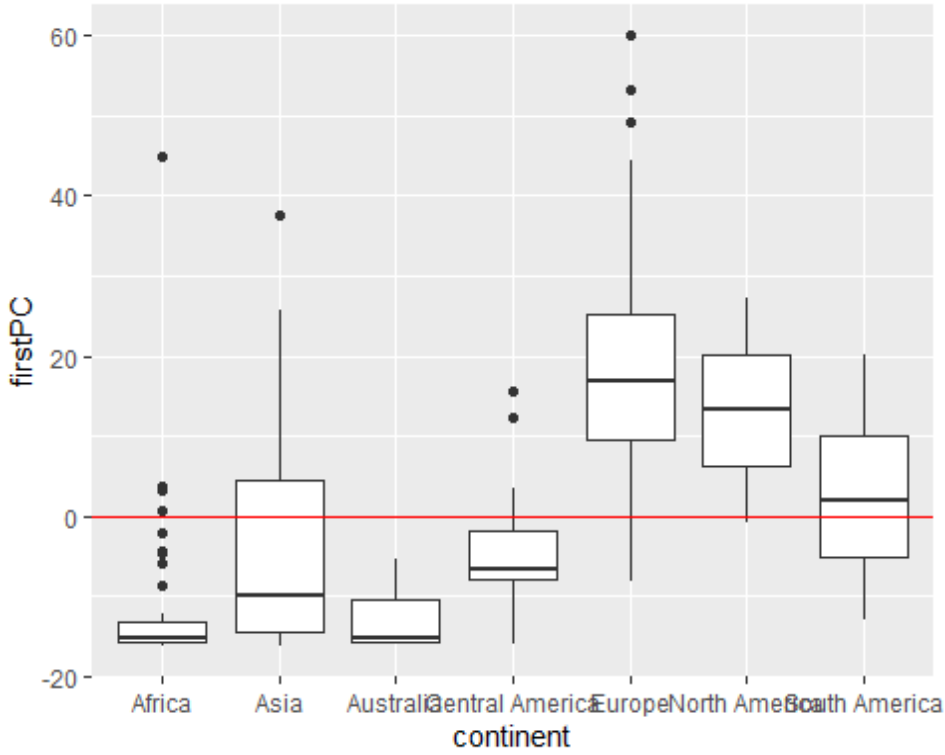
```
##              Df Sum Sq Mean Sq F value Pr(>F)
## continent     6  25062    4177   27.62 <2e-16 ***
## Residuals   172  26008     151
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There are significant differences among continents(p-value small).

```
library(ggplot2)
```

```
## Warning: il pacchetto 'ggplot2' è stato creato con R versione 4.2.3
```

```
ggplot(data = anova_data,
aes(continent,firstPC))+geom_boxplot()+geom_hline(yintercept = 0, col = "red")
```



Africa, Asia, Australia and Central America have mainly negative scores. Europe, North America, South America have more positive scores.