Assignment 7

Sebastian Veuskens, Desmond Reynolds

Exercise 1

Load data

```
library(fpc)
library(poLCA)

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

```
##              MORALG            CARESG             KNOWG            LEADG
## 1    3 Not too well  1 Extremely well      2 Quite well     2 Quite well
## 2 4 Not well at all    3 Not too well 4 Not well at all   3 Not too well
## 3  1 Extremely well      2 Quite well      2 Quite well 1 Extremely well
## 4      2 Quite well      2 Quite well      2 Quite well     2 Quite well
## 5      2 Quite well 4 Not well at all      2 Quite well   3 Not too well
## 6      2 Quite well    3 Not too well    3 Not too well     2 Quite well
##           DISHONG       INTELG           MORALB           CARESB          KNOWB
## 1 3 Not too well 2 Quite well 1 Extremely well 1 Extremely well   2 Quite well
## 2   2 Quite well 2 Quite well             <NA>             <NA>   2 Quite well
## 3 3 Not too well 2 Quite well     2 Quite well     2 Quite well   2 Quite well
## 4   2 Quite well 2 Quite well     2 Quite well   3 Not too well   2 Quite well
## 5   2 Quite well 2 Quite well 1 Extremely well 1 Extremely well   2 Quite well
## 6   2 Quite well 2 Quite well     2 Quite well             <NA> 3 Not too well
##             LEADB           DISHONB           INTELB
## 1   2 Quite well 4 Not well at all     2 Quite well
## 2 3 Not too well              <NA> 1 Extremely well
## 3 3 Not too well    3 Not too well     2 Quite well
## 4   2 Quite well    3 Not too well 1 Extremely well
## 5   2 Quite well    3 Not too well     2 Quite well
## 6   2 Quite well      2 Quite well     2 Quite well
```

```
electionwithna <- election12
for (i in 1:12){
    levels(electionwithna[,i]) <- c(levels(election12[,i]),"NA")
    electionwithna[is.na(election12[,i]),i] <- "NA"
}

election14 <- election[14:15]
election14[3:14] <- electionwithna
for (i in 1:2) {
    election14[is.na(election14[,i]),i] <- mean(election14[,i], na.rm=T)
}
```

Prepare MDS plot

```
library(cluster)
library(smacof)

dist_election <- daisy(electionwithna, metric="gower")

mds_election <- mds(dist_election, ndim=2)
mds_election$stress
```

```
## [1] 0.3096474
```

```
show_mds <- function(clustering, title) {
    plot(mds_election$conf, col=clustering$cluster, main=title)
}
```

a)

```
f2a <- cbind(MORALG,CARESG,KNOWG,LEADG,DISHONG,INTELG,
             MORALB,CARESB,KNOWB,LEADB,DISHONB,INTELB)~1

clus_a <- poLCA(f2a, electionwithna, nclass=3, graphs=TRUE)
```

```
## 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:           0.4440       0.3695         0.0437            0.0291 0.1137
## class 2:           0.1044       0.6185         0.2141            0.0207 0.0423
## class 3:           0.1687       0.3443         0.2303            0.2076 0.0491
##
## $CARESG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.3678       0.3883         0.0799            0.0726 0.0913
## class 2:           0.0253       0.5275         0.3527            0.0649 0.0296
## class 3:           0.0762       0.2308         0.3514            0.2939 0.0477
##
## $KNOWG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.5214       0.3544         0.0284            0.0278 0.0681
## class 2:           0.0618       0.8047         0.1258            0.0046 0.0031
## class 3:           0.2114       0.4615         0.2209            0.0802 0.0260
##
## $LEADG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.3472       0.4215         0.0906            0.0419 0.0988
## class 2:           0.0213       0.5210         0.3969            0.0374 0.0233
## class 3:           0.0684       0.2297         0.3942            0.2746 0.0330
##
## $DISHONG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.0321       0.0714         0.2353            0.4993 0.1619
## class 2:           0.0299       0.1852         0.5145            0.2365 0.0340
## class 3:           0.1907       0.2884         0.2669            0.1872 0.0668
##
## $INTELG
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.5317       0.3556         0.0302            0.0340 0.0486
## class 2:           0.0650       0.8058         0.1072            0.0080 0.0140
## class 3:           0.2618       0.4544         0.1836            0.0798 0.0204
##
## $MORALB
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.0842       0.2912         0.2809            0.1362 0.2074
## class 2:           0.0453       0.6663         0.2103            0.0150 0.0630
## class 3:           0.5293       0.4165         0.0294            0.0107 0.0141
##
## $CARESB
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.0113       0.0928         0.3500            0.4289 0.1170
## class 2:           0.0037       0.4230         0.4364            0.1036 0.0334
## class 3:           0.2992       0.5659         0.0981            0.0223 0.0145
##
## $KNOWB
##          1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:           0.0854       0.3073         0.3072             0.204 0.0962
## class 2:           0.0122       0.7105         0.2537             0.014 0.0097
```

```
## class 3:            0.4394       0.5227         0.0356             0.000 0.0023
##
## $LEADB
##           1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:            0.0278       0.2357         0.3568            0.2484 0.1313
## class 2:            0.0320       0.6581         0.2590            0.0192 0.0316
## class 3:            0.4663       0.4992         0.0239            0.0051 0.0055
##
## $DISHONB
##           1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:            0.0785       0.2482         0.2750            0.1662 0.2322
## class 2:            0.0130       0.1469         0.5488            0.2318 0.0595
## class 3:            0.0277       0.0741         0.2180            0.6425 0.0377
##
## $INTELB
##           1 Extremely well 2 Quite well 3 Not too well 4 Not well at all     NA
## class 1:            0.1190       0.3630         0.2609            0.1615 0.0956
## class 2:            0.0334       0.7333         0.1989            0.0168 0.0175
## class 3:            0.4803       0.4908         0.0212            0.0014 0.0063
##
## Estimated class population shares
##  0.3387 0.3885 0.2727
##
## Predicted class memberships (by modal posterior prob.)
##  0.3333 0.395 0.2717
##
## =========================================================
## Fit for 3 latent classes:
## =========================================================
## number of observations: 1785
## number of estimated parameters: 146
## residual degrees of freedom: 1639
## maximum log-likelihood: -25972.59
##
## AIC(3): 52237.18
## BIC(3): 53038.31
## G^2(3): 25636.59 (Likelihood ratio/deviance statistic)
## X^2(3): 42945467129 (Chi-square goodness of fit)
##
```

```
clus_a$cluster <- clus_a$predclass
```

b)

```
flex_clus <- 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
## Nonoptimal or repeated fit found in run  3
## Nonoptimal or repeated 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
## k=  3  new best fit found in run  10
## Nonoptimal or repeated fit found in run  11
## Nonoptimal or repeated fit found in run  12
## Nonoptimal or repeated 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=  52863.7
```

```
clus_b <- list()
clus_b$cluster <- flex_clus$flexout[[3]]@cluster
```

c)

Hierarchical clustering with average linkage

```
clus_c = list()
clus_c$cluster <- cutree(hclust(dist_election, method="average"), k=3)
```

d)

```
set.seed(12345)
optimal_k <- flexmixedruns(electionwithna, continuous = 0, discrete=12, n.cluster=1:10)$optim
alk
```

```
## k=  1  new best fit found in run  1
## k=  1  BIC=  57445.61
## k=  2  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## k=  2  new best fit found in run  3
## Nonoptimal or repeated 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
## k=  2  new best 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
## Nonoptimal or repeated 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=  2  BIC=  54295.51
## k=  3  new best fit found in run  1
## k=  3  new best fit found in run  2
## Nonoptimal or repeated 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=  52863.71
## k=  4  new best fit found in run  1
## k=  4  new best fit found in run  2
## k=  4  new best fit found in run  3
## Nonoptimal or repeated fit found in run  4
## k=  4  new best fit found in run  5
## Nonoptimal or repeated fit found in run  6
## Nonoptimal or repeated fit found in run  7
## k=  4  new best 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
```

```
## k=  4  new best fit found in run  12
## k=  4  new best fit found in run  13
## k=  4  new best 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=  4  BIC=  51682.54
## k=  5  new best fit found in run  1
## k=  5  new best fit found in run  2
## k=  5  new best fit found in run  3
## Nonoptimal or repeated fit found in run  4
## Nonoptimal or repeated fit found in run  5
## Nonoptimal or repeated fit found in run  6
## k=  5  new best fit found in run  7
## Nonoptimal or repeated fit found in run  8
## k=  5  new best 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
## Nonoptimal or repeated 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=  5  BIC=  51158.82
## k=  6  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## Nonoptimal or repeated fit found in run  3
## Nonoptimal or repeated fit found in run  4
## Nonoptimal or repeated fit found in run  5
## Nonoptimal or repeated fit found in run  6
## k=  6  new best fit found in run  7
## k=  6  new best 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
## Nonoptimal or repeated 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
## k=  6  new best 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=  6  BIC=  50858.43
## k=  7  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## Nonoptimal or repeated fit found in run  3
## Nonoptimal or repeated fit found in run  4
```

```
## Nonoptimal or repeated fit found in run  5
## Nonoptimal or repeated fit found in run  6
## k=  7  new best 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
## k=  7  new best fit found in run  11
## Nonoptimal or repeated fit found in run  12
## Nonoptimal or repeated fit found in run  13
## Nonoptimal or repeated fit found in run  14
## Nonoptimal or repeated fit found in run  15
## k=  7  new best fit found in run  16
## Nonoptimal or repeated fit found in run  17
## Nonoptimal or repeated fit found in run  18
## k=  7  new best fit found in run  19
## k=  7  new best fit found in run  20
## k=  7  BIC=  50761.99
## k=  8  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## Nonoptimal or repeated fit found in run  3
## k=  8  new best fit found in run  4
## k=  8  new best 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
## Nonoptimal or repeated 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=  8  BIC=  50745.88
## k=  9  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## k=  9  new best fit found in run  3
## Nonoptimal or repeated 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
## Nonoptimal or repeated 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=  9  BIC=  50813.49
## k=  10  new best fit found in run  1
## k=  10  new best fit found in run  2
## k=  10  new best fit found in run  3
## Nonoptimal or repeated fit found in run  4
## k=  10  new best 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
## Nonoptimal or repeated 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=  10  BIC=  50912.88
```

```
clus_d <- list()
flex_clus <- flexmixedruns(electionwithna, continuous = 0, discrete = 12, n.cluster=optimal_
k)
```

```
## k=  8  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## Nonoptimal or repeated fit found in run  3
## Nonoptimal or repeated fit found in run  4
## Nonoptimal or repeated fit found in run  5
## k=  8  new best 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
## Nonoptimal or repeated 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=  8  BIC=  50743.89
```

```
clus_d$cluster <- flex_clus$flexout[[optimal_k]]@cluster
clus_d$n.class <- optimal_k
optimal_k
```

```
## [1] 8
```

The best number of clusters here is 8 clusters.

e)

```
set.seed(12345)
election14_numeric <- sapply(election14, unclass)
clus_e <- flexmixedruns(election14_numeric, continuous=2, discrete=12, n.cluster=1:10)
```

```
## k=  1  new best fit found in run  1
## k=  1  BIC=  79391.33
## k=  2  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## k=  2  new best fit found in run  3
## Nonoptimal or repeated 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
## k=  2  new best 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
## Nonoptimal or repeated 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=  2  BIC=  76256.63
## 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
## Nonoptimal or repeated 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
## k=  3  new best fit found in run  10
## k=  3  new best fit found in run  11
## Nonoptimal or repeated fit found in run  12
## Nonoptimal or repeated 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
## k=  3  new best fit found in run  18
## Nonoptimal or repeated fit found in run  19
## Nonoptimal or repeated fit found in run  20
## k=  3  BIC=  74828.7
## k=  4  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## Nonoptimal or repeated fit found in run  3
## Nonoptimal or repeated 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
## Nonoptimal or repeated 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=  4  BIC=  73635.65
## k=  5  new best fit found in run  1
## k=  5  new best fit found in run  2
## k=  5  new best fit found in run  3
## Nonoptimal or repeated fit found in run  4
## k=  5  new best 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
## Nonoptimal or repeated 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=  5  BIC=  73114.55
## k=  6  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## Nonoptimal or repeated fit found in run  3
## Nonoptimal or repeated fit found in run  4
## Nonoptimal or repeated fit found in run  5
## Nonoptimal or repeated fit found in run  6
## k=  6  new best 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
## k=  6  new best fit found in run  11
## Nonoptimal or repeated fit found in run  12
## Nonoptimal or repeated 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=  6  BIC=  72831.04
## k=  7  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## Nonoptimal or repeated fit found in run  3
## Nonoptimal or repeated fit found in run  4
```

```
## Nonoptimal or repeated fit found in run  5
## k=  7  new best 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
## k=  7  new best fit found in run  10
## k=  7  new best fit found in run  11
## k=  7  new best fit found in run  12
## Nonoptimal or repeated 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
## k=  7  new best fit found in run  20
## k=  7  BIC=  72765.21
## k=  8  new best fit found in run  1
## k=  8  new best fit found in run  2
## Nonoptimal or repeated fit found in run  3
## k=  8  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
## k=  8  new best 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
## Nonoptimal or repeated 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=  8  BIC=  72764.94
## k=  9  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## k=  9  new best fit found in run  3
## k=  9  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
## Nonoptimal or repeated fit found in run  13
## Nonoptimal or repeated fit found in run  14
## k=  9  new best fit found in run  15
## Nonoptimal or repeated fit found in run  16
## Nonoptimal or repeated fit found in run  17
## k=  9  new best fit found in run  18
```

```
## Nonoptimal or repeated fit found in run  19
## Nonoptimal or repeated fit found in run  20
## k=  9  BIC=  72818.13
## k=  10  new best fit found in run  1
## Nonoptimal or repeated fit found in run  2
## Nonoptimal or repeated fit found in run  3
## k=  10  new best fit found in run  4
## Nonoptimal or repeated fit found in run  5
## k=  10  new best 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
## Nonoptimal or repeated 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=  10  BIC=  72927.52
```

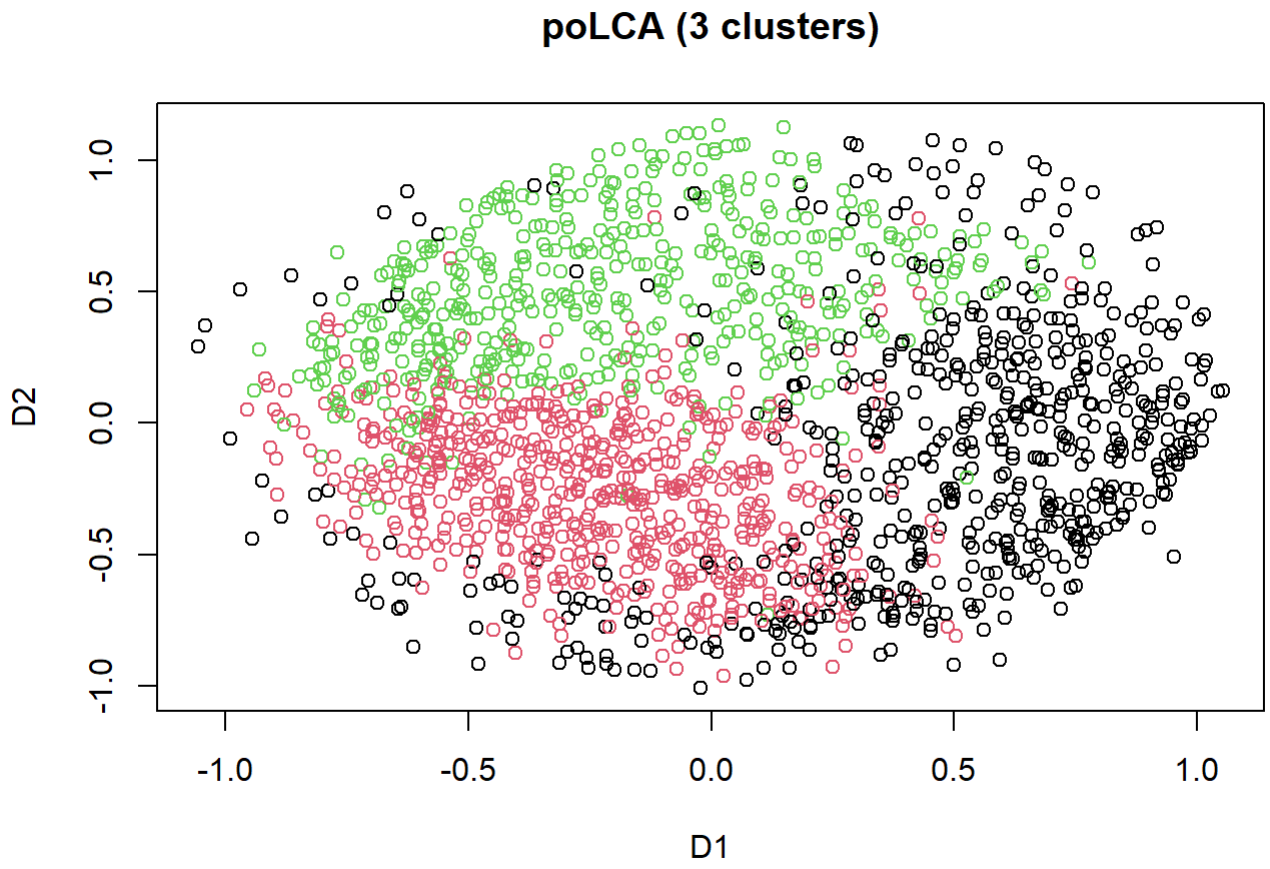
```
clus_e$cluster <- flex_clus$flexout[[flex_clus$optimalk]]@cluster
flex_clus$optimalk
```

```
## [1] 8
```

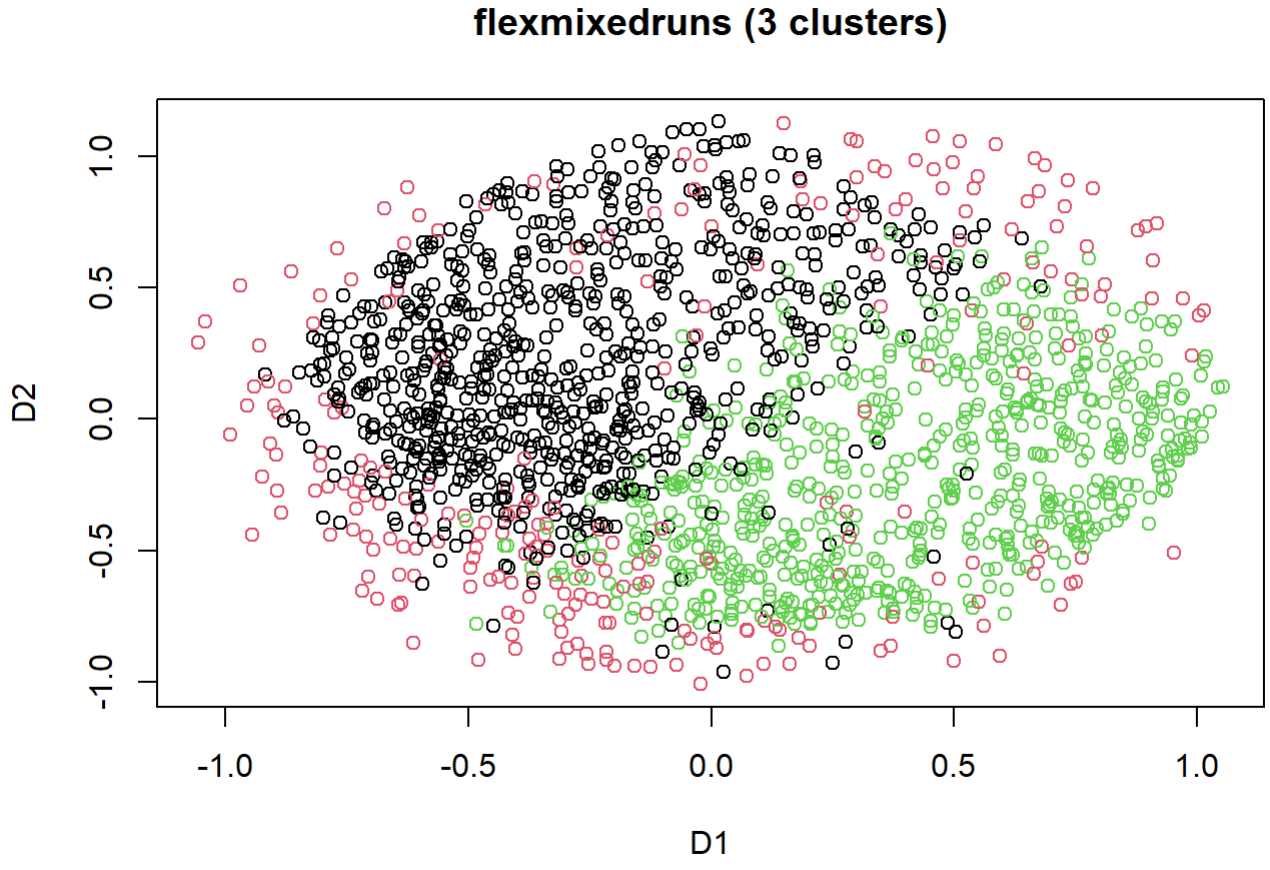
The best number of clusters is in this case 8 clusters.

Compare clusterings

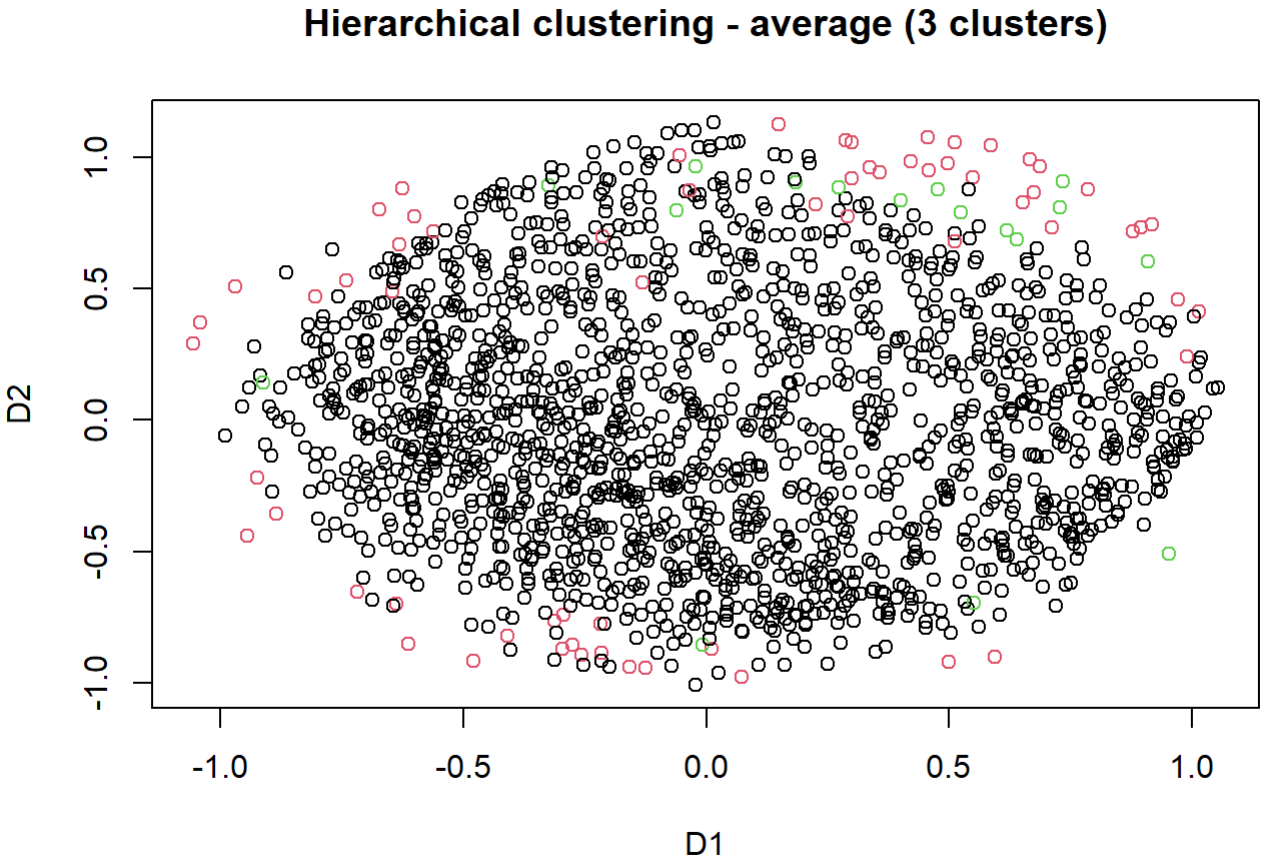
```
show_mds(clus_a, "poLCA (3 clusters)")
```



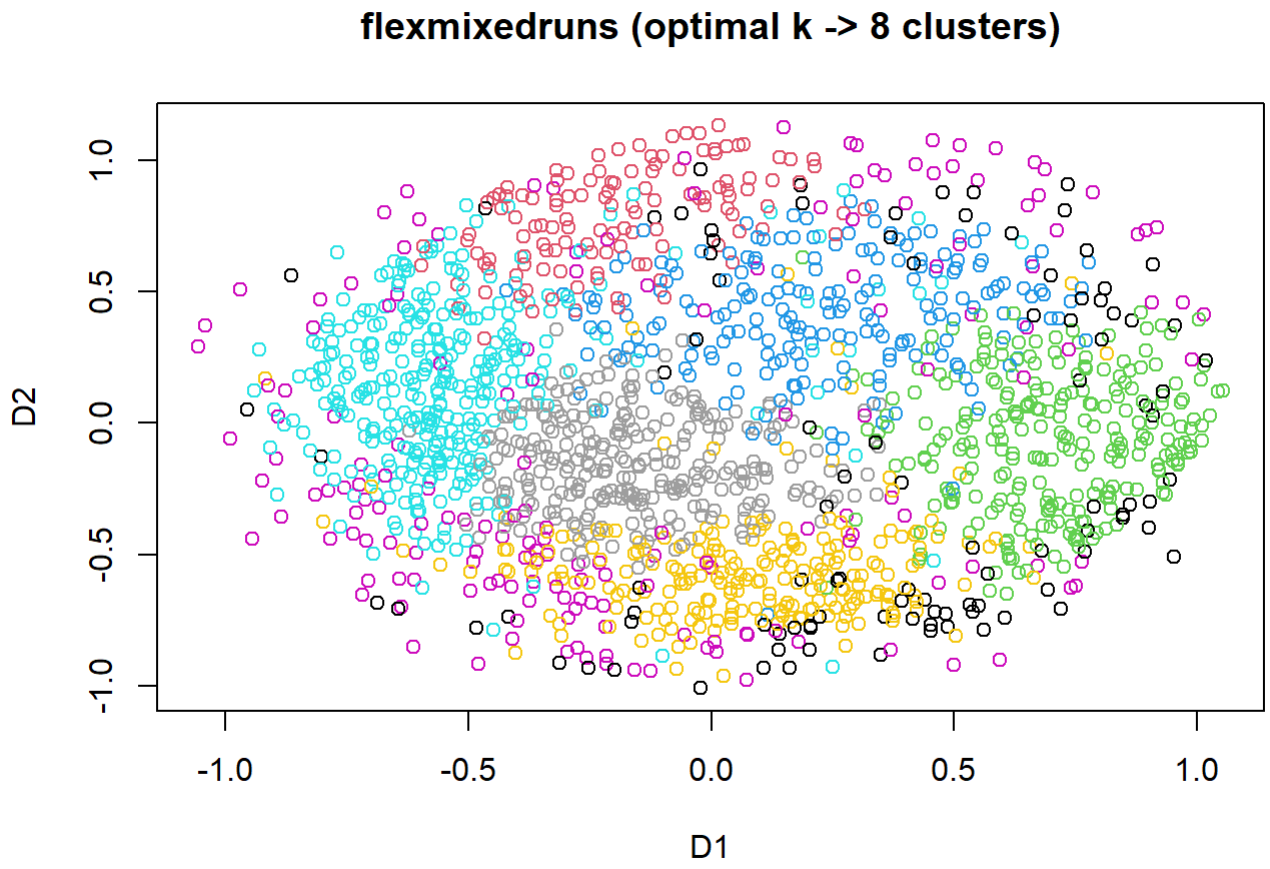

```
show_mds(clus_b, "flexmixedruns (3 clusters)")
```



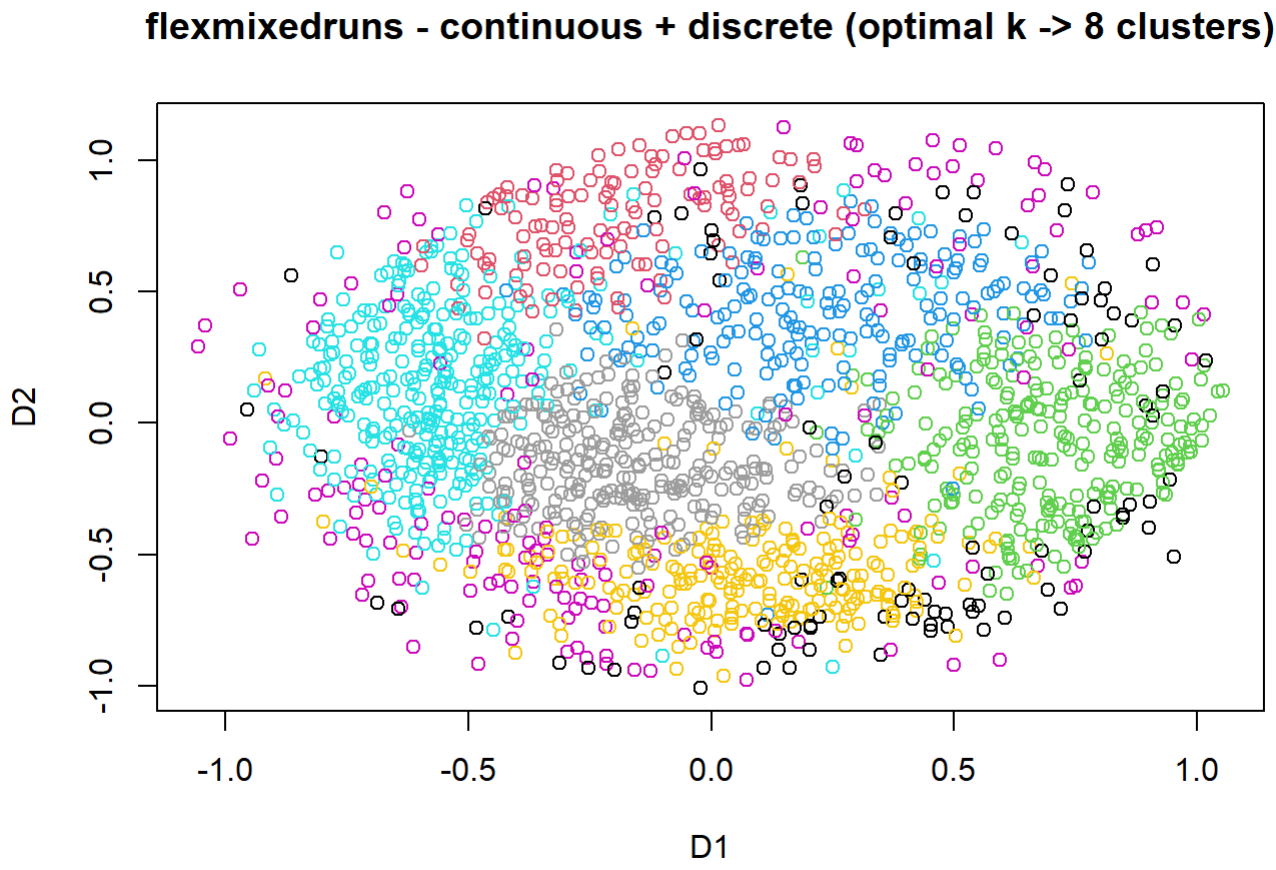
```
show_mds(clus_c, "Hierarchical clustering - average (3 clusters)")
```




```
show_mds(clus_d, "flexmixedruns (optimal k -> 8 clusters)")
```




```
show_mds(clus_e, "flexmixedruns - continuous + discrete (optimal k -> 8 clusters)")
```



From the mds plot it is hard to tell which clustering seems more convincing. No clear distinctive clusters regarding both separation or density are visible. However, the clearest distinction and close clusters are generated by the poLCA clustering method, although the green cluster is not a clear cluster, since it spans into the realm of other clusters, according to the mds.

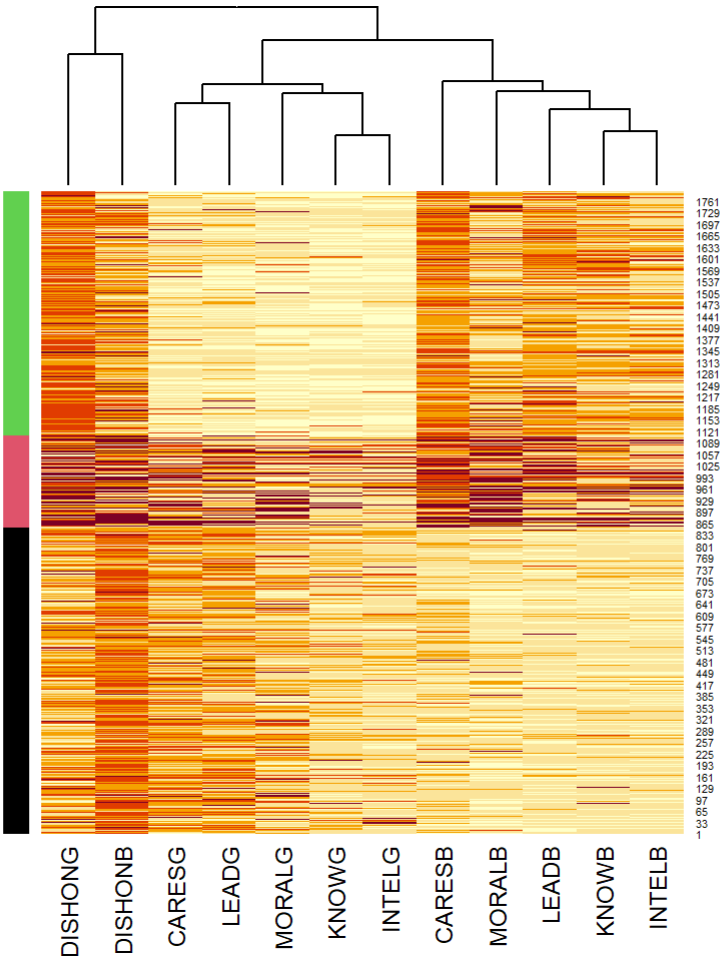
In addition, both flexmixedruns clusterings with flexible clusters behave very similarly and show a fairly good separation for some of the clusters. But overall, the clusters are noisier than for the poLCA.

Hierarchical clustering rarely catches any significant cluster, rather it detects some outliers and assigns them to different groups. All other observations are assigned to the same main cluster.

Exercise 2

Create clustering for clus_b (flexmixedruns -> 3 clusters)

```
# Create numeric matrix for display in heatmap
election_numeric <- sapply(electionwithna[,], unclass)

# col_dist_election <- daisy(as.data.frame(t(electionwithna)), metric="gower")
col_dist_election <- daisy(t(election_numeric), metric="gower", type=list(factor=c(1:1785)))
#
heatmap(election_numeric[order(clus_b$cluster),],
        Rowv=NA, Colv=as.dendrogram(hclust(col_dist_election, method="average")),
        RowSideColors=palette()[clus_b$cluster][order(clus_b$cluster)],
        scale="none")
```



We think the clustering clearly differentiates between inherently different observations. Each group shows a distinctive pattern. While the black cluster is constantly higher than the other two clusters, the other two clusters show different columns in which they have high or low values, respectively. Thus, the clustering seems

like a good clustering to me.

Local independence is hard to spot. But from the given plot, no clear violation of local independence is visible. A violation of this assumption would be characterized by two columns showing similar patterns of the observations within each cluster. This is not the case.

Exercise 3

```
# a )
(2**5)*(3**3)*(5**2) - 1
```

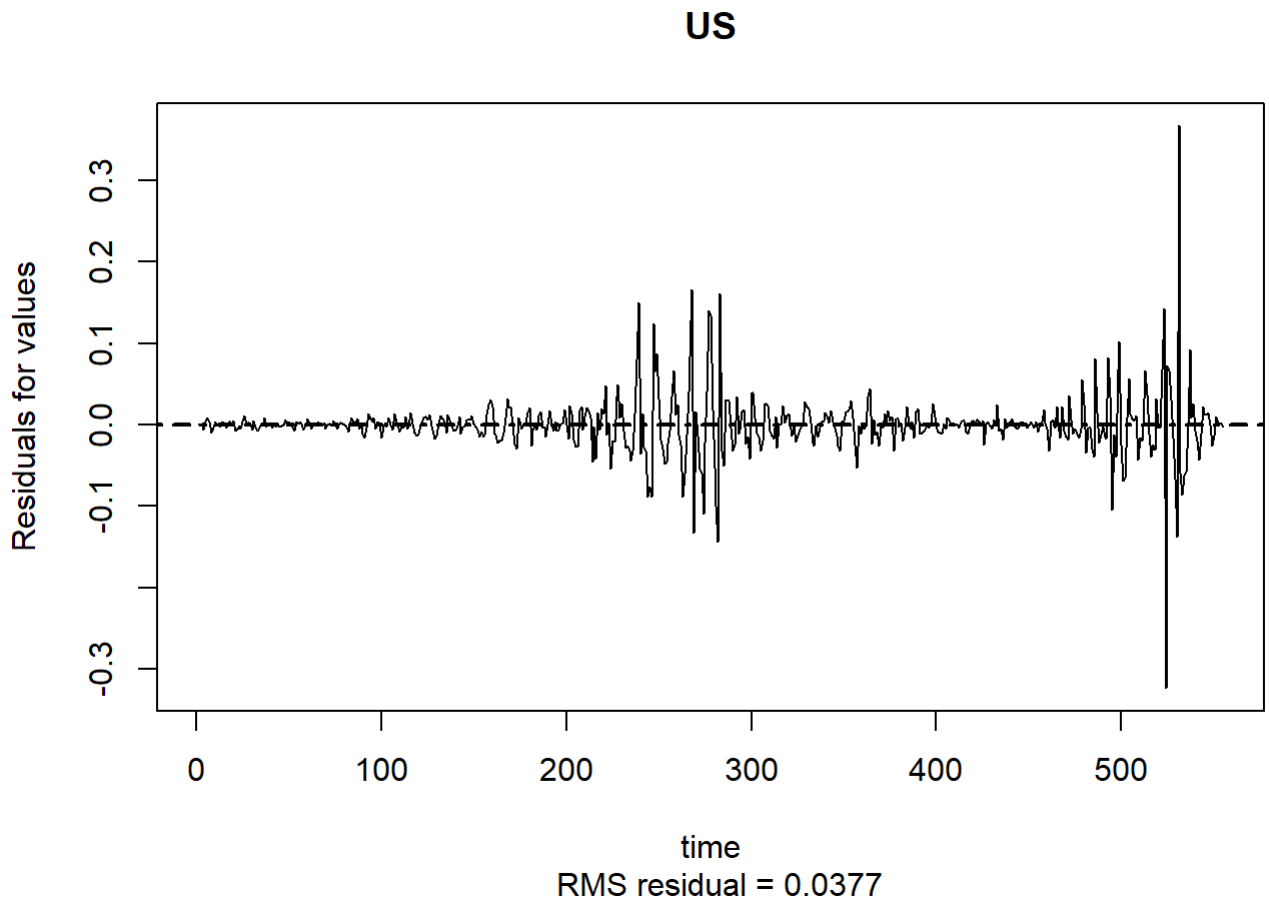
```
## [1] 21599
```

```
# b)
4 * ((2 - 1) * 5 + (3 - 1) * 3 + (5 - 1) * 2 + 1) - 1
```

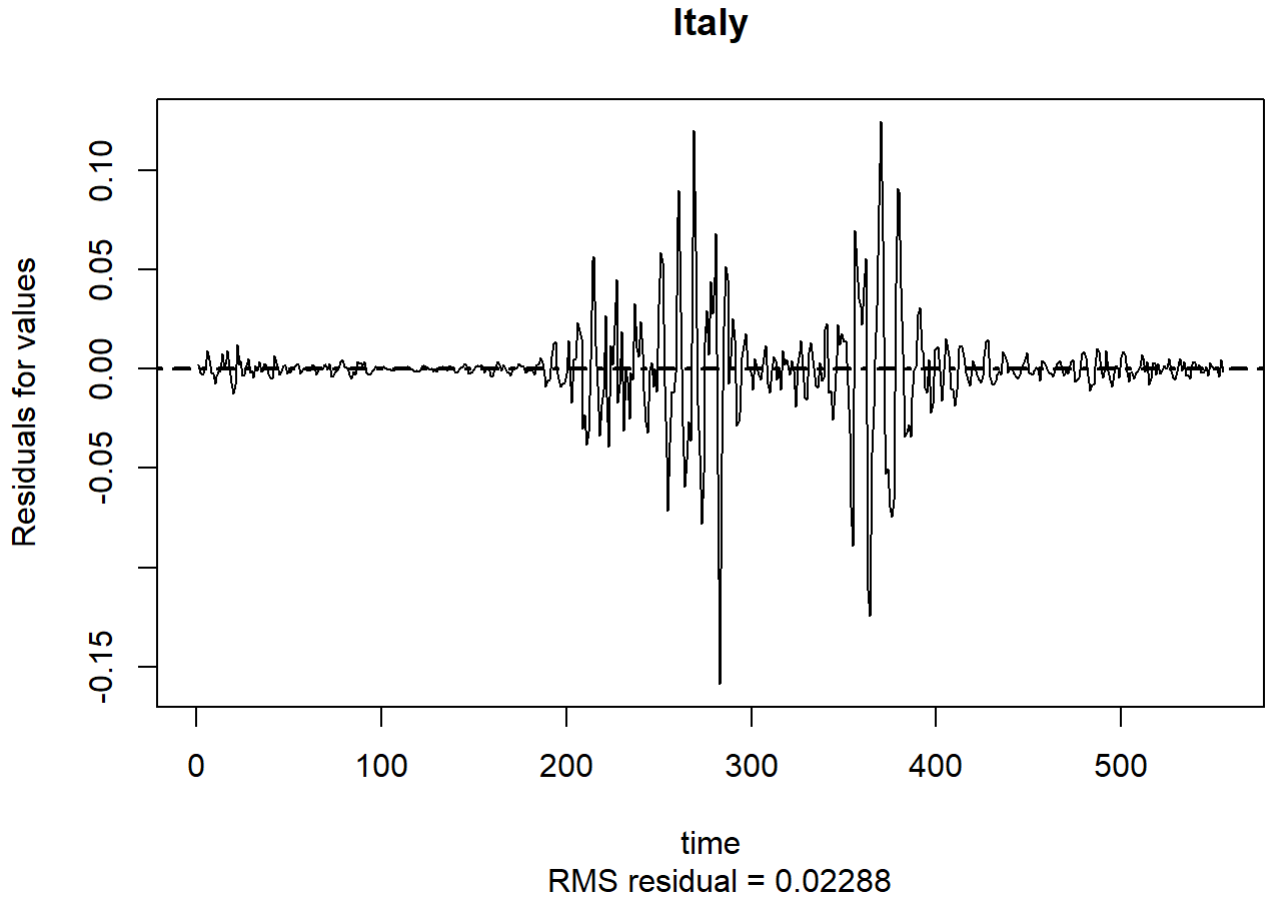
```
## [1] 79
```

Exercise 4

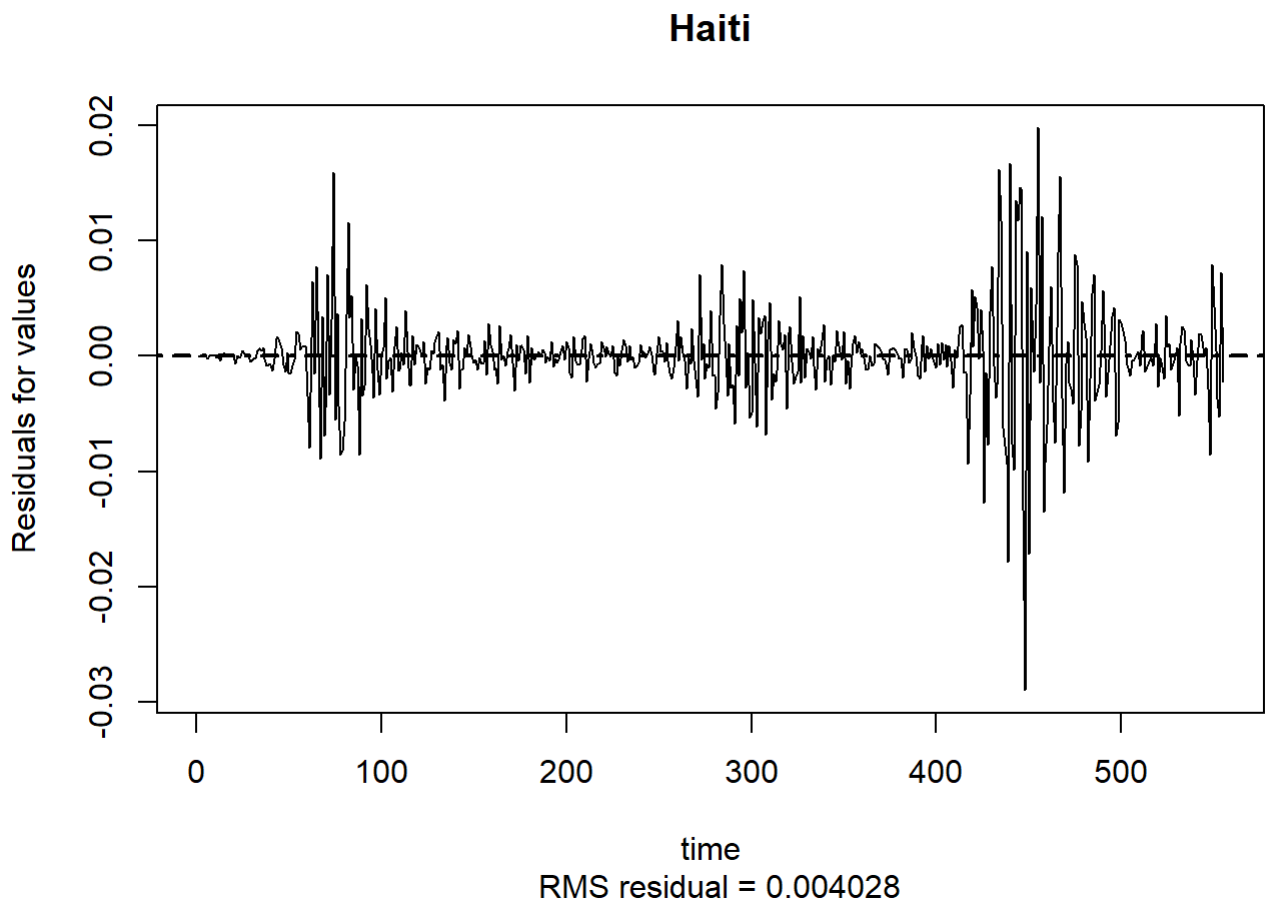
```
library(fda)
# Load data
covid21 <- read.table("data/covid2021.dat")
covid21v <- as.matrix(covid21[,5:559])

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

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




```
plotfit.fd(t(covid21v), 1:555, fdcovid100, index=79, cex.pch=0.5, residual=T)
```



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



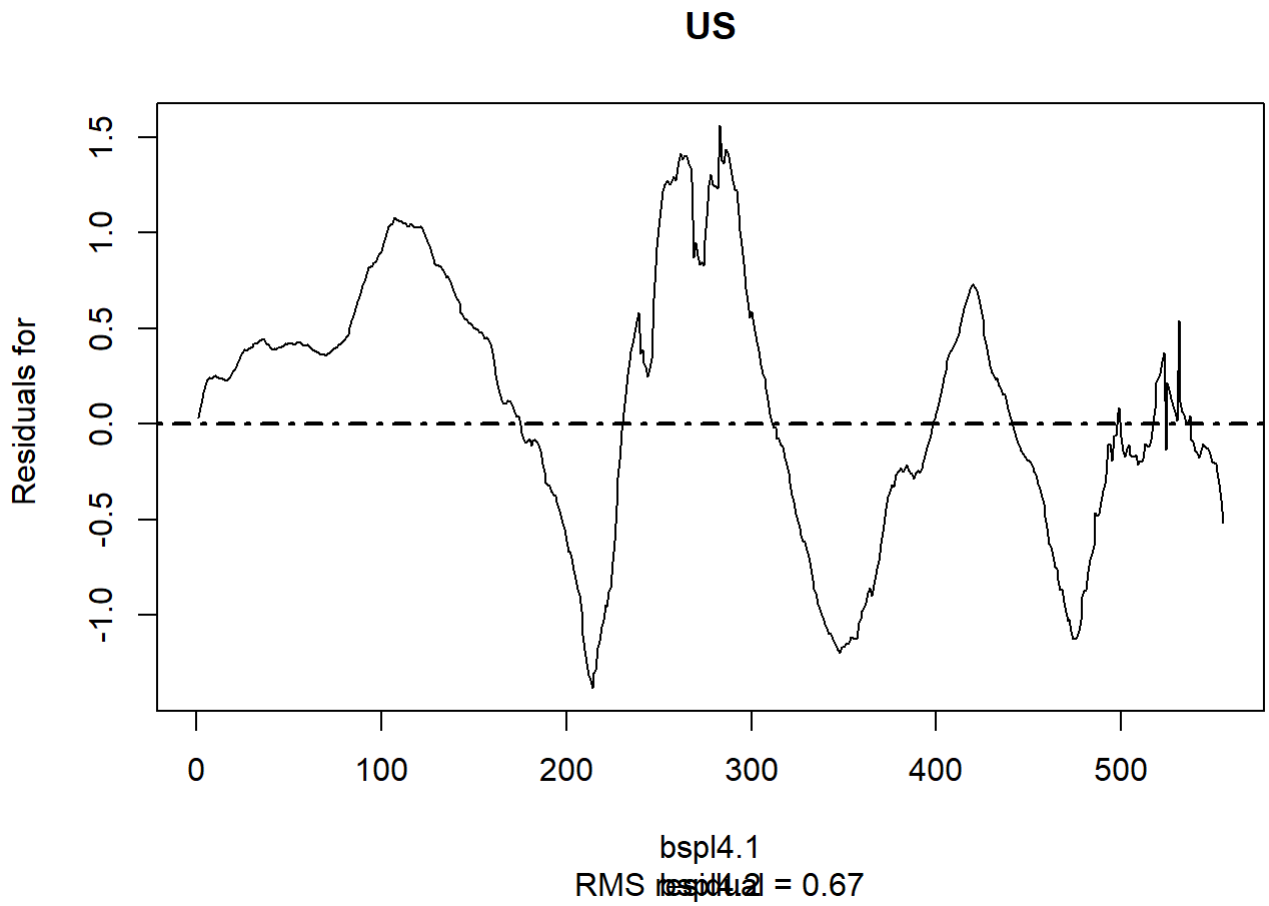

```
# Principal Component basis
covidpca <- pca.fd(fdcovid100, nharm = 5)
covidpca$varprop
```

```
## [1] 0.40871917 0.18555649 0.11717077 0.05581147 0.04606967
```

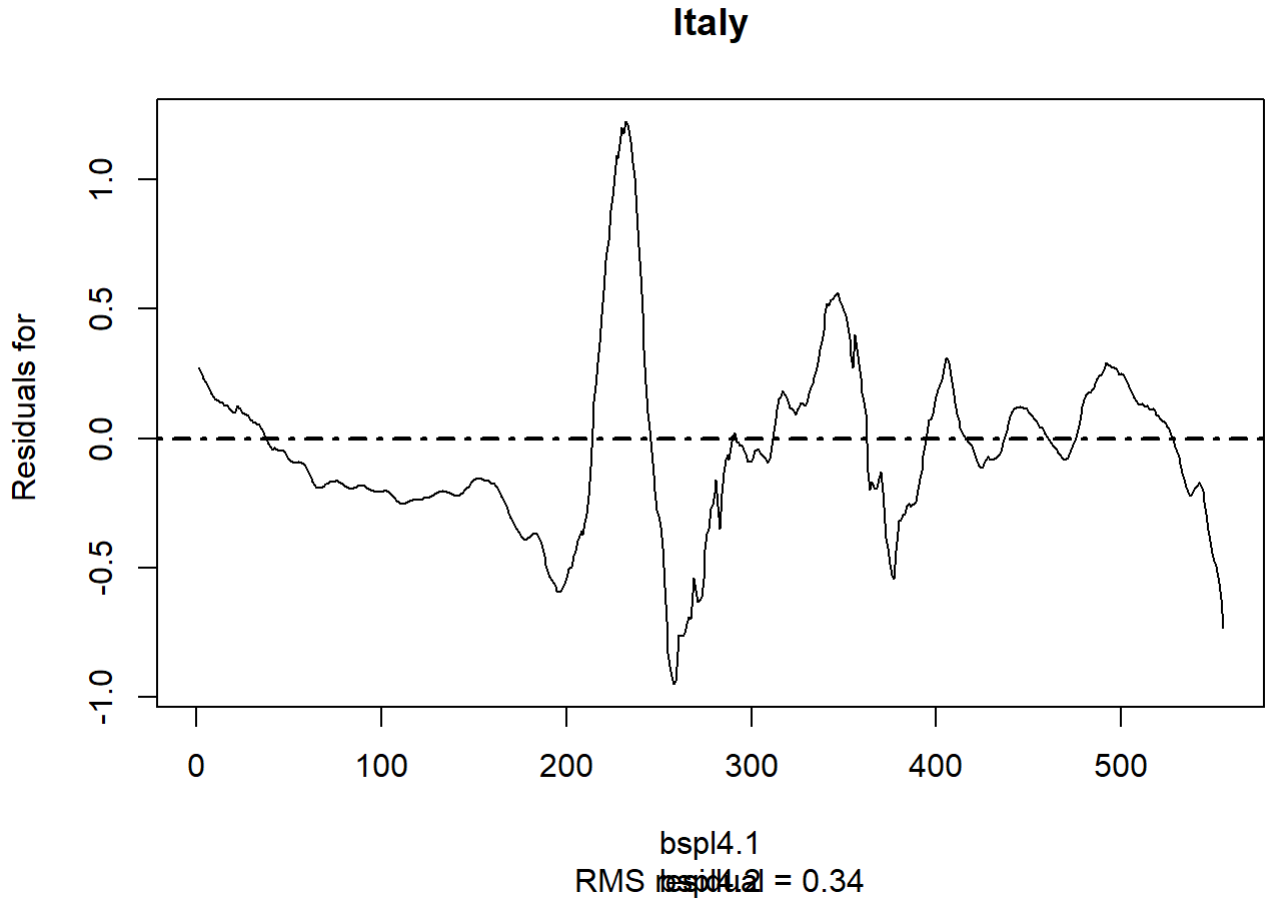
```
cumsum(covidpca$varprop)
```

```
## [1] 0.4087192 0.5942757 0.7114464 0.7672579 0.8133276
```

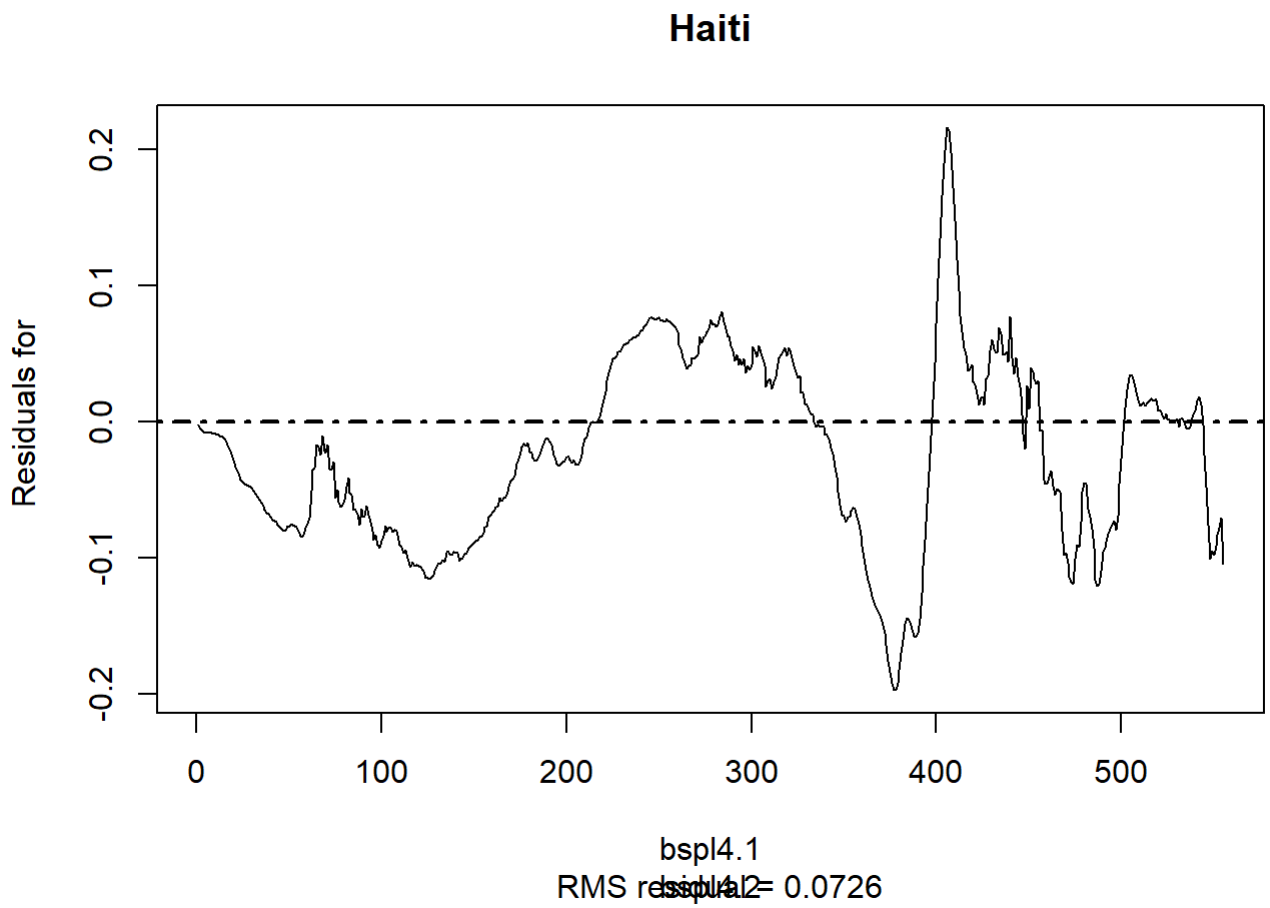
```
mcovid <- mean.fd(fdcovid100)

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=164,cex.pch=0.5, residual=T)
```




```
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)
```



For the B-spline basis, all approximations are fairly close to the original timeseries of the three different countries. The range of absolute maximum deviation from the original data ranges from *0.03 (Haiti)* to *0.3 (US)*. Anyway, for evey country there is some areas where the approximation systematically differs from the original data for consecutive time steps. For example, for Italy, between step *200* to *300*: While the approximation is correct on average, the approximation is seems to miss some error noise. This probably happens in areas where a lot of Covid cases are present and thus the numbers fluctuate more.

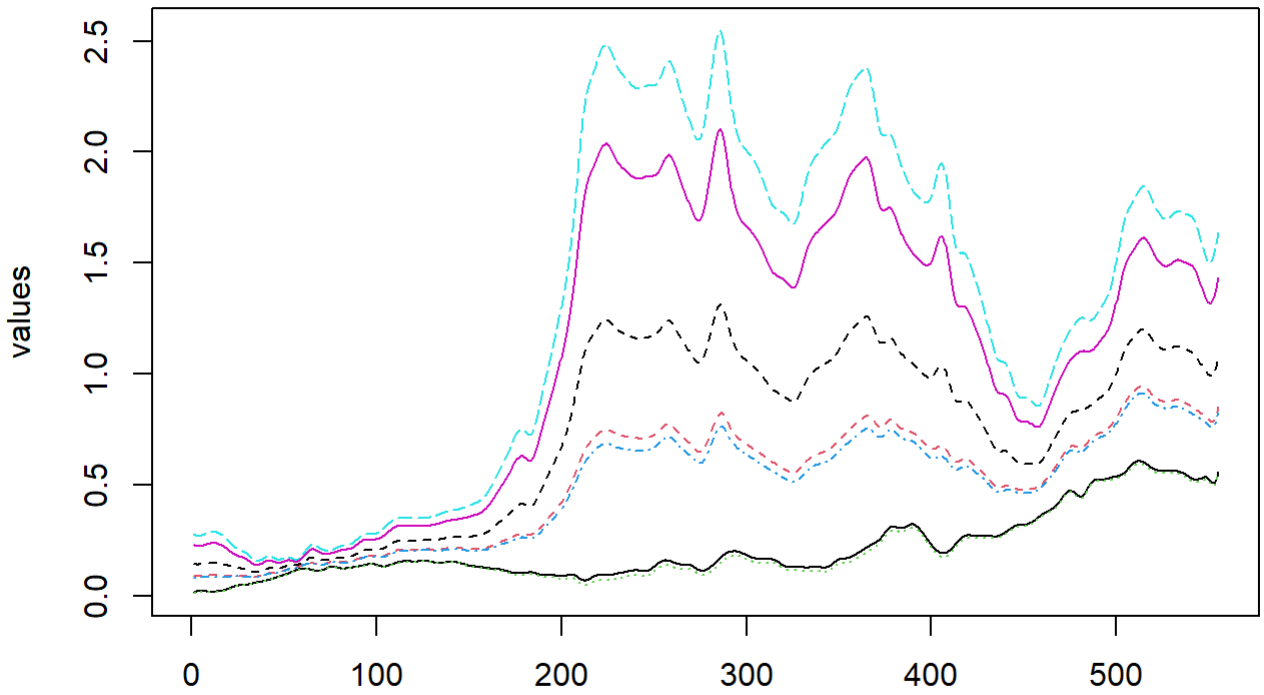
For the PCA basis with 5 components, on the other hand, relatively much higher deviation can be observed from the residual plot. There the range of absolute maximum deviation from the original data ranges from *0.2 (Haiti)* to *1.0 (Italy and US)*. This indicates a great loss by a 5-dimensional PCA representation. Characteristic patterns of the single countries are not displayed appropriately.

Exercise 5

```
library(dplyr)
covidpca1 <- pca.fd(fdcovid100, nharm = 1)
ncontinent <- as.numeric(as.factor(covid21$continent))
con_names <- levels(as.factor(covid21$continent))

con_scores_all <- data.frame(scores = covidpca1$scores, continent = as.factor(ncontinent))

res.aov <- aov(scores ~ continent, data = con_scores_all)
# Significant difference between continent mean scores
summary(res.aov)
```

```
##              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
```

```
con_scores <- group_by(con_scores_all, continent) %>%
    summarise(
        mean = mean(scores)
    )

covidpca1 <- pca.fd(fdcovid100, nharm = 1)
covidpca1$scores <- con_scores$mean
covidpca1approx <- covidpca1$harmonics
i <- 1
pcacoefi1 <- covidpca1$harmonics$coefs %*% con_scores$mean[i]+mcovid$coefs
covidpca1approx$coefs <- pcacoefi1
for (i in 2:length(con_scores$mean)) {
    pcacoefi1 <- covidpca1$harmonics$coefs %*% con_scores$mean[i]+mcovid$coefs
    covidpca1approx$coefs <- cbind(covidpca1approx$coefs, pcacoefi1)
}

attributes(covidpca1approx$coefs)$dimnames[[2]] <- con_names
plot(covidpca1approx)
```




```
## [1] "done"
```

The continents differ significantly from each other, regarding the scores of the first principal component.

The plot indicates that in general, a great amount of the data is explained by the first PC for Europe and, to a lesser amount, in North America. This indicates that in these continents, the greatest variation took place.

On the other hand, for Africa and Australia, very low negative mean scores are reported. This indicates a contrary dynamic in comparison to the dynamics of the first component of the PCs.

Values close to zero (Asia, Central America and South America) indicate a low variation from the general mean function.