```r
# Convexity-based clustering application using the Lilly
# 6-week longitudinal data set HCAF.
# Classify prozac treated subjects to be either:
# 1. placebo responders
# 2. drug responders
# 3. placebo non-responders
# 4. drug non-responders.


setwd('/Users/yaolanqiu/Desktop/NYU/rotation/Rotation2/Week3')
library(lme4)
library(splines)
library(fda)  # Use Ramsay's code to obtain design matrices for various
library(mgcv)
source("cvxcluster-0513.R")
# Read in the Lilly data set hcaf.dat
# Column 1: subject ID
# Column 2: treatment indicator (0=placebo, 1=fluoxetine, 2=Imipramine)
# Column 3: y = HRSD (outcome variable over time)
# Column 4: age
# Column 5: Baseline CGI score
# Column 6: t1 = week
#   note: t1=0 is randomization time
# Column 7: t2 = week^2 (for quadratic fit)
setwd('/Users/yaolanqiu/Desktop/NYU/rotation/Rotation2/Week3')

dat <- read.table("hcaf.dat", header=T)

dim(dat) # 3364 7
```

```
## [1] 3364    7
```

```r
length(unique(dat$subj)) # 543
```

```
## [1] 543
```

```r
length(unique(dat$t1)) # 7
```

```
## [1] 7
```

```r
# > unique(dat$t1)
# [1] 0 1 2 3 4 5 6

# Define the matrix A to convert quadratic curves to an
# orthogonal polynomial basis: If X is usual design matrix,
# then X%*%A is the design matrix for the orthogonal polynomial

t <- as.matrix(0:6) # pt = the order of time points
ni <- length(t) # 7
X = cbind(matrix(1, length(t), 1), t, t^2)
Xtpo <- X
tbar = mean(t) # 3
Xtpo[, 2] = X[, 2] - tbar
Xtpo[, 3] = (t - tbar)^2 - (ni^2 - 1) / 12
c0 <- sqrt(sum(Xtpo[,1]^2))
c1 <- sqrt(sum(Xtpo[,2]^2))
```

```
c2 <- sqrt(sum(Xtpo[,3]^2))

c0
```

```
## [1] 2.645751
```

```
c1
```

```
## [1] 5.291503
```

```
c2
```

```
## [1] 9.165151
```

```
Xtpo[,1] = Xtpo[,1] / c0
Xtpo[,2] = Xtpo[,2] / c1
Xtpo[,3] = Xtpo[,3] / c2
A <- matrix(0,3,3) # A = transformation matrix
A[1, 1] <- 1 / c0
A[1, 2] <- - tbar / c1
A[2, 2] <- 1 / c1
A[1, 3] <- (tbar^2 - (ni^2 - 1) / 12) / c2
A[2, 3] <- -2*tbar / c2
A[3, 3] <- 1 / c2
A
```

```
##           [,1]       [,2]       [,3]
## [1,] 0.3779645 -0.5669467  0.5455447
## [2,] 0.0000000  0.1889822 -0.6546537
## [3,] 0.0000000  0.0000000  0.1091089
```

```
dim(X)
```

```
## [1] 7 3
```

```
p <- dim(X)[2]
p
```

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

```
placebo <- NULL
prozac <- NULL
dat = subset(dat, trt != 2) # We are not using the imi treatment here

dim(dat) # 2209 7
```

```
## [1] 2209    7
```

```
length(unique(dat$subj)) # 358
```

```
## [1] 358
```

```
length(unique(dat$t1)) # 7
```

```
## [1] 7
```

```
for (jt in unique(dat$trt)){ # fit lme for each arm
  dati <- dat[dat$trt == jt,]
  fit1 <- lmer(y ~ t1 + I(t1^2) + (t1+I(t1^2)|subj), data = dati, REML = FALSE)
  D <- as.matrix(VarCorr(fit1)$subj) # Covariance matrix for random effects
  D <- D[1:p, 1:p]
```

```r
  beta <- as.matrix(fixef(fit1))
  sigma <- attr(VarCorr(fit1), "sc")
  bis <- as.matrix(coef(fit1)$subj) %*% t(solve(A))

  responder <- NULL # record subjects that are responders or not
  age <- NULL
  BaselineCGI <- NULL
  for (isubj in unique(dati$subj)){
    datisubj <- dati[dati$subj ==isubj,]
    responder <- rbind(responder, datisubj$responder[1])
    age <- rbind(age, datisubj$age[1])
    BaselineCGI <- rbind(BaselineCGI, datisubj$BaselineCGI[1])
  }

  if (jt == 0){
    placebo$n <- length(unique(dati$subj))
    placebo$dat <- dati
    placebo$D <- D
    placebo$beta <- beta
    placebo$sigma <- sigma
    placebo$bis <- bis
    placebo$responder <- responder
    placebo$age <- age
    placebo$BaselineCGI <- BaselineCGI
    placebo$fit <- fit1
  }
  if (jt == 1){
    prozac$n <- length(unique(dati$subj))
    prozac$dat <- dati
    prozac$D <- D
    prozac$beta <- beta
    prozac$sigma <- sigma
    prozac$bis <- bis
    prozac$responder <- responder
    prozac$age <- age
    prozac$BaselineCGI <- BaselineCGI
    prozac$fit <- fit1
  }
}

beta
```

```
##                    [,1]
## (Intercept) 23.6891978
## t1          -4.2926488
## I(t1^2)      0.3689679
```

```r
D
```

```
##             (Intercept)         t1    I(t1^2)
## (Intercept)   8.1890822  0.3375841 -0.2183801
## t1            0.3375841  7.8743792 -1.0483529
## I(t1^2)      -0.2183801 -1.0483529  0.1616528
```

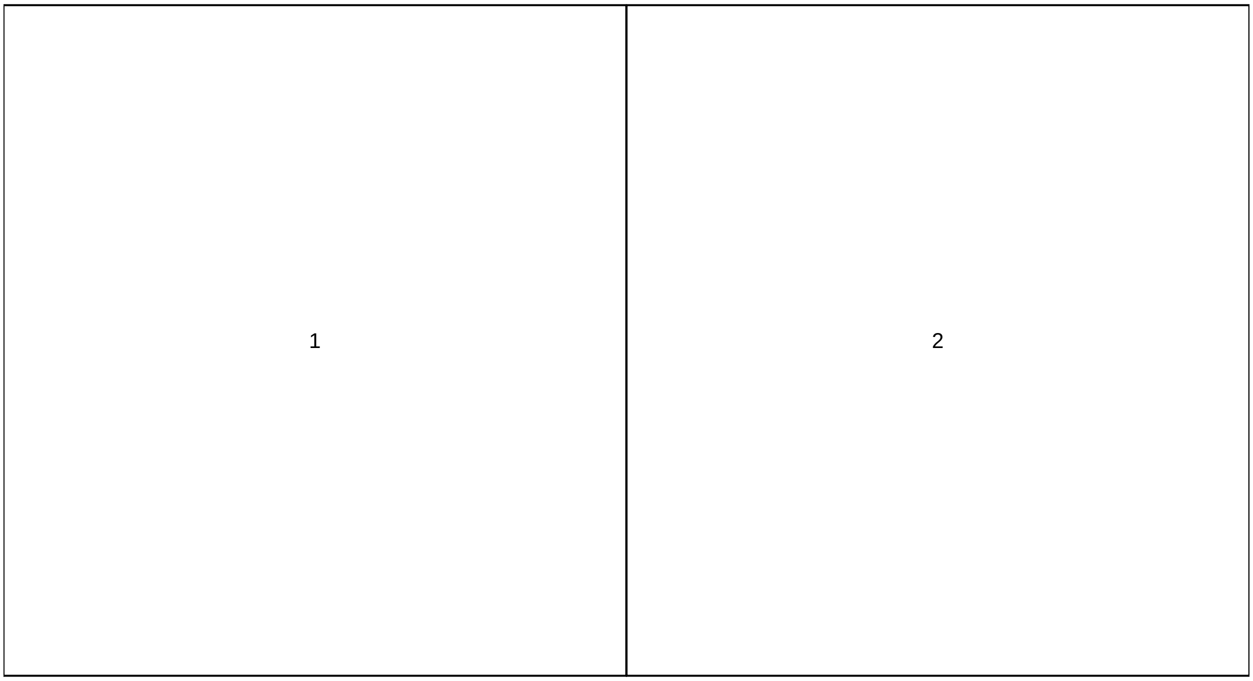
```
sigma
```

```
## [1] 3.427945
```

```
head(bis)
```

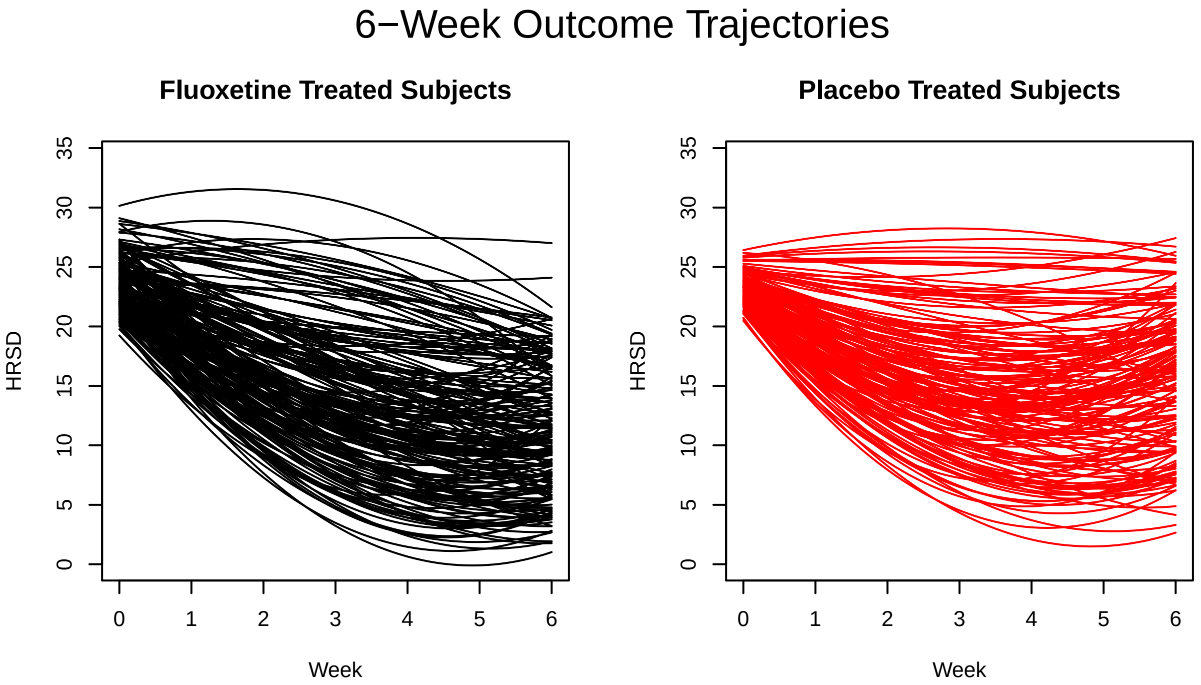
```
##          [,1]       [,2]       [,3]
## 2500 53.78619  -6.599916  0.2541448
## 2509 31.01795 -11.725623  5.8386620
## 2511 57.97193  -4.121040 -1.2735774
## 2514 41.33329  -7.703816  5.5288367
## 2515 42.69898 -10.058038  3.8271343
## 2524 28.90097 -13.738185  8.0152515
```

```
# plot the trajectories
#########FIGURE 3###################
nf <- layout(matrix(c(0, 0, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 0, 0),
                    9, 2, byrow = TRUE))
layout.show(nf)
```

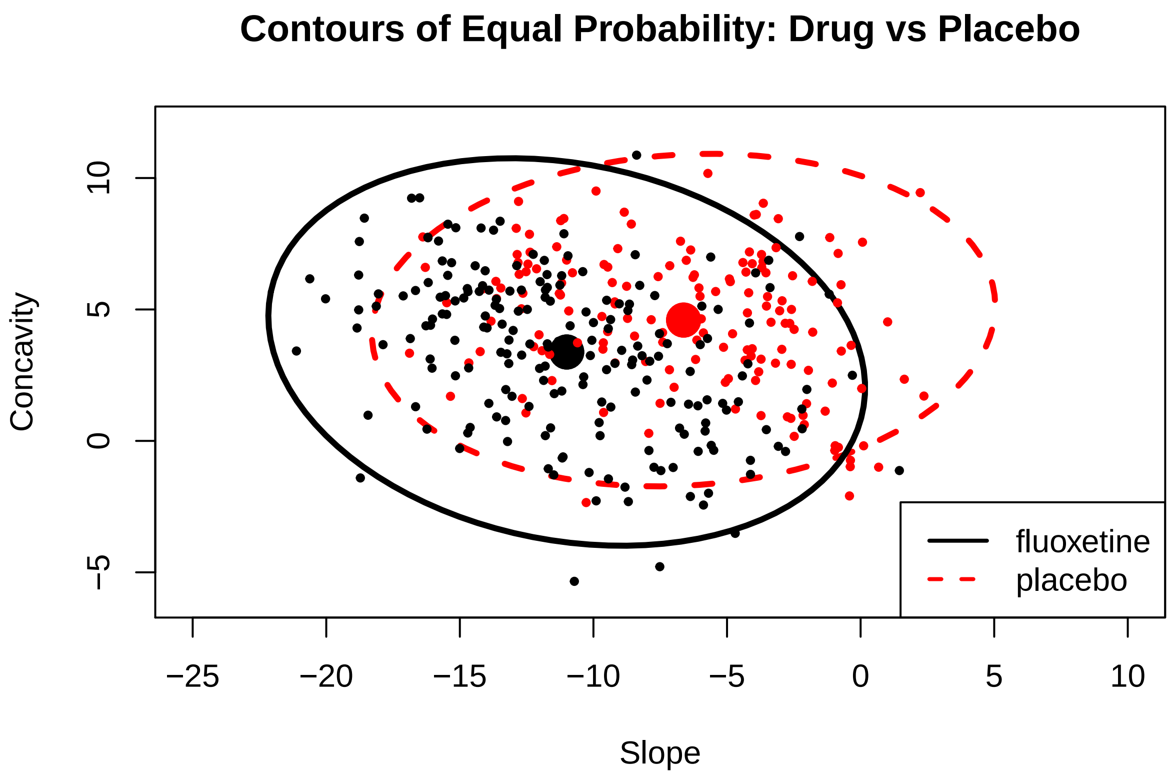


```
tplot = seq(0, 6, by = .1)
plot(t, t * 5.7, type = "n", main="Fluoxetine Treated Subjects",
     xlab = "Week", ylab = "HRSD")
for (i in 1:prozac$n){
  bi = A%*%as.matrix(prozac$bis[i, 1:3])
  lines(tplot, bi[1,1] + bi[2,1] * tplot + bi[3 , 1] * tplot^2, col = 1)
}
plot(t, t*5.7, type = "n", main = "Placebo Treated Subjects",
     xlab = "Week", ylab = "HRSD")
for (i in 1:placebo$n){
  bi = A %*% as.matrix(placebo$bis[i, ])
  lines(tplot, bi[1, 1] + bi[2, 1] * tplot+bi[3, 1] * tplot^2, col=2)
}
```

```
mtext("6-Week Outcome Trajectories", outer=TRUE, side=3, cex=1.25, line=-4)
```

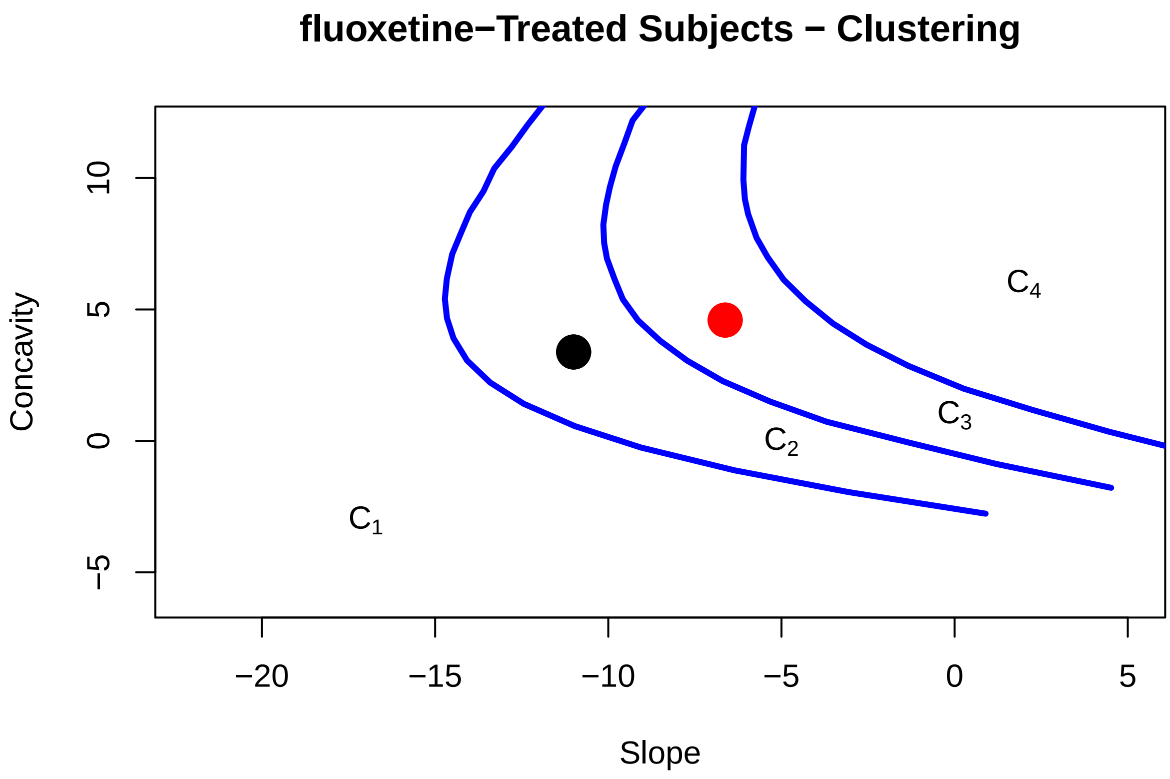


```
#####

# Simulate data from both arms and define estimated density functions
beta <- placebo$beta
D <- solve(A) %*% placebo$D %*% t(solve(A))
xbar <- solve(A) %*% beta
pbobeta = as.numeric(xbar[2:3,])
mu1 = pbobeta
pboD = D[2:3,2:3]

beta <- prozac$beta
D <- solve(A) %*% prozac$D %*% t(solve(A))
xbar <- solve(A) %*% beta
prozbeta = as.numeric(xbar[2:3, ])
mu2 = prozbeta
prozD = D[2:3, 2:3]

data = as.data.frame( rbind(placebo$bis[, 2:3], prozac$bis[, 2:3]) )
names(data) = c("slope", "concavity")
data$group = c(rep(1, placebo$n), rep(2, prozac$n))

# Here we are using the function by defining the mean and covariance matrix
# of the two populations,
# because we would like to use the mean and covariance estimated
# by the mixed effect model
# We could also use the cvxcluster function by input the dataset and
# grouping variables such as
# cvxcluster(data0 = data, by = "group", k = 4)
k = 4
p1 = cvxcluster(miu1 = pbobeta, cov1 = pboD, miu2 = prozbeta,
                cov2 = prozD, k = 4, nsim = 50000, niter = 20)
x = p1$xsim #simulated observations
u = p1$bound #threshold of the clusters


p2 = cvxcluster(miu1 = pbobeta, cov1 = pboD, miu2 = prozbeta,
                cov2 = prozD, k = 4, nsim = 50, niter = 1)


# draw confidence ellipsoids
############FIGURE 4#################
par(mfrow=c(1,1))
nellipse = 100  # number of points for drawing ellipses
c = 4  # amount to stretch ellipses
epbo = eigen(pboD)
eproz = eigen(prozD)
theta = seq(0,2*pi, length.out = nellipse)
ellip1 = cbind(cos(theta), sin(theta))
ellip2 = ellip1
ellip1 = ellip1 %*% sqrt(diag(c*epbo$values)) %*% t(epbo$vectors)
ellip2 = ellip2 %*% sqrt(diag(c*eproz$values)) %*% t(eproz$vectors)
ellip1 = ellip1 + t(matrix(mu1, 2, nellipse))
ellip2 = ellip2 + t(matrix(mu2, 2, nellipse))

plot(x[,2:3], type = "n", xlab = "Slope", ylab = "Concavity",
     xlim = c(-25, 10), ylim = c(-6,12),
     main = "Contours of Equal Probability: Drug vs Placebo")
mus = rbind(t(mu1), t(mu2))
points( rbind(mus[1, ], mus[1, ]), cex = 2.3, pch = 19, col = 2)
points( rbind(mus[2, ], mus[2, ]), cex = 2.3, pch = 19, col = 1)
lines(ellip1, col = 2, lty = 2, lwd = 3)
lines(ellip2, lwd = 3)
legend("bottomright", c("fluoxetine", "placebo"),
       lwd = c(2,2), col = c(1,2), lty = c(1,2))
points(placebo$bis[,2:3], col = 2, pch = 19, cex = 0.5)
points(prozac$bis[,2:3], col = 1, pch = 19, cex = 0.5)
```



```
#####
```

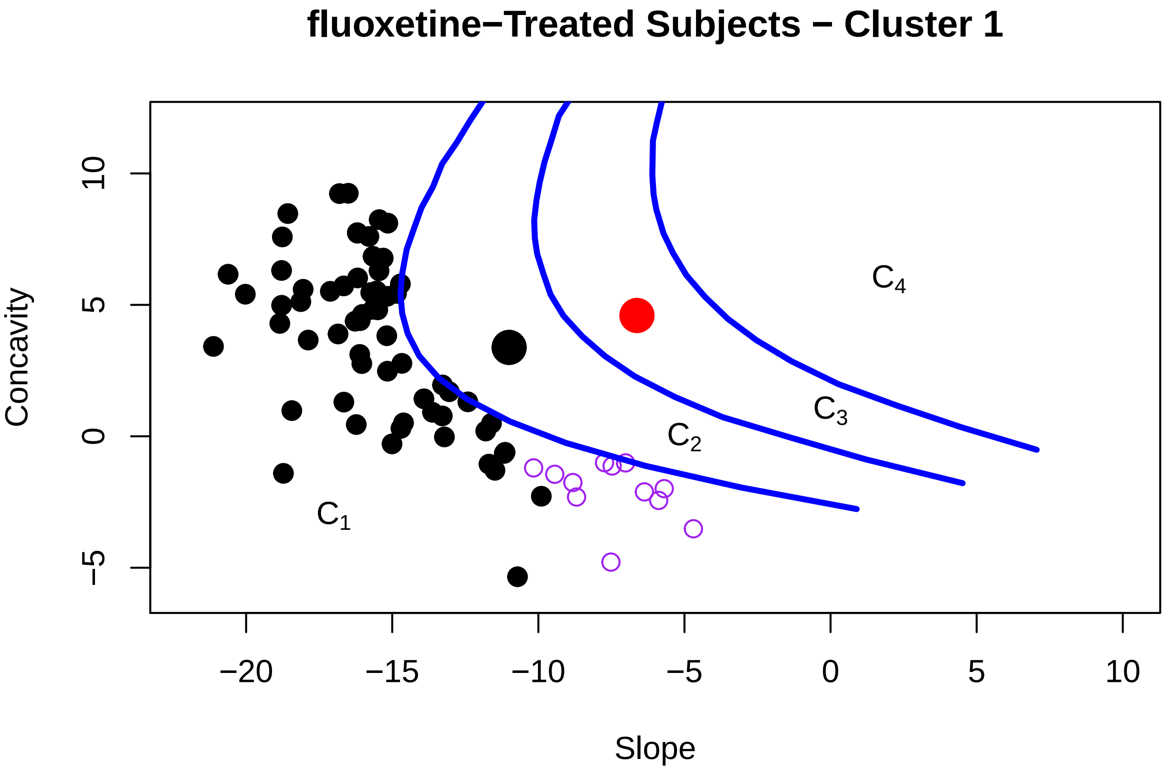
```
##### Plot boundaries of the partition
################FIGURE 5#################
plot(x[, 2:3], type = "n", xlab = "Slope", ylab = "Concavity",
     xlim = c(-22,5), ylim = c(-6,12),
     main = "fluoxetine-Treated Subjects - Clustering")

for (j in 1:(k-1)){
  Bcurvej = NULL
  epsilon = 1
  nc = 25
  Bj = as.matrix(x[x$cluster == j, 1:2])
  By = sort(Bj[, 2])
  By = seq(By[1], By[length(By)], length.out = nc)
  for (ic in 1:nc){
    Bslice = Bj[(Bj[, 2] > By[ic] - epsilon) & (Bj[, 2] < By[ic] + epsilon),]
    if (is.matrix(Bslice) == T){
      Bcurvej = rbind(Bcurvej, Bslice[Bslice[, 1] == min(Bslice[, 1]), ])
    }
  }
  lines(Bcurvej[,1], Bcurvej[,2], lwd = 3, col = 4)
}
points( rbind(mus[1,], mus[1,]), cex = 2.3, pch = 19, col = 2)
points( rbind(mus[2,], mus[2,]), cex = 2.3, pch = 19, col = 1)
text(-17, -3, expression(C[1]))
text(-5, 0, expression(C[2]))
text(0, 1, expression(C[3]))
text(2, 6, expression(C[4]))
```



```
#####

# classify actual subjects
data$lambda = NA
data$cluster = NA
for (i in 1:length(data$group)){
  data$lambda[i] = lambda(data[i, 1:2], d = 2,
                          f1 = list(miu=pbobeta, cov=pboD),
                          f2 = list(miu=prozbeta, cov=prozD),
                          pi1 = 0.5, pi2 = 0.5)
  if (!is.na(data$lambda[i])) {
    data$cluster[i] = 1
    for (j in 1:(k-1)){
      if (data$lambda[i] >= u[j]) data$cluster[i] = j + 1
    }
  }
}
```

```
# plot cluster 4, responder and non-responder
############FIGURE 6############################
indx = (data$cluster == 4)
drugclass4 = indx[(placebo$n+1): (placebo$n+prozac$n)]

plot(x[,1:2], type = "n", xlab = "Slope", ylab = "Concavity",
     xlim = c(-22,10), ylim = c(-6,12),
     main = "fluoxetine-Treated Subjects - Cluster 1")

points(prozac$bis[drugclass4, 2:3], col="purple", pch=1, cex=1.2)
points(prozac$bis[drugclass4 & prozac$responder==1,2:3],
       col="black", cex=1.3, pch=19)# Plot boundaries of the partition
```

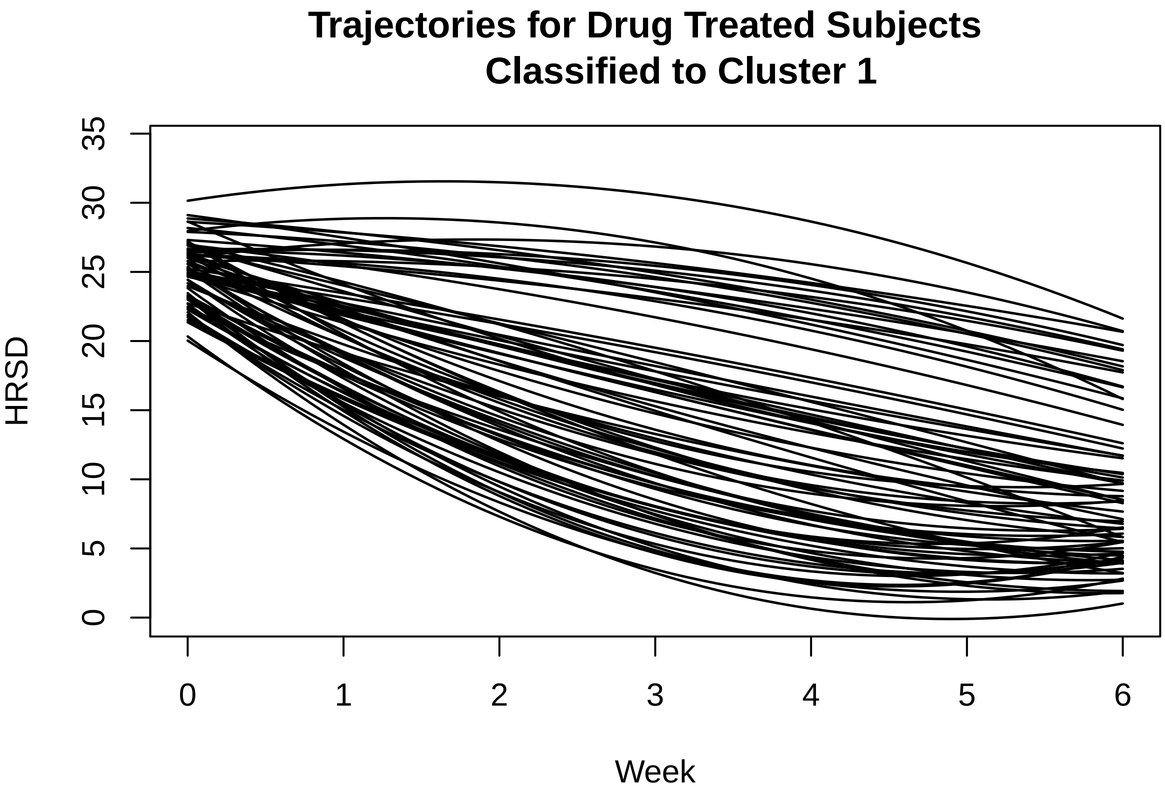
```
for (j in 1:(k-1)){
  Bcurvej = NULL
  epsilon = 1
  nc = 25
  Bj = as.matrix(x[x$cluster == j, 1:2])
  By = sort(Bj[, 2])
  By = seq(By[1], By[length(By)], length.out = nc)
  for (ic in 1:nc){
    Bslice = Bj[(Bj[, 2] > By[ic] - epsilon) & (Bj[, 2] < By[ic] + epsilon),]
    if (is.matrix(Bslice) == T){
      Bcurvej = rbind(Bcurvej, Bslice[Bslice[,1] == min(Bslice[, 1]), ])
    }
  }
  lines(Bcurvej[, 1], Bcurvej[, 2], lwd = 3, col = 4)
}
points( rbind(mus[1, ], mus[1, ]), cex = 2.3, pch = 19, col = 2)
points( rbind(mus[2, ], mus[2, ]), cex = 2.3, pch = 19, col = 1)
text(-17, -3, expression(C[1]))
text(-5, 0, expression(C[2]))
text(0, 1, expression(C[3]))
text(2, 6, expression(C[4]))
```



```
######
```

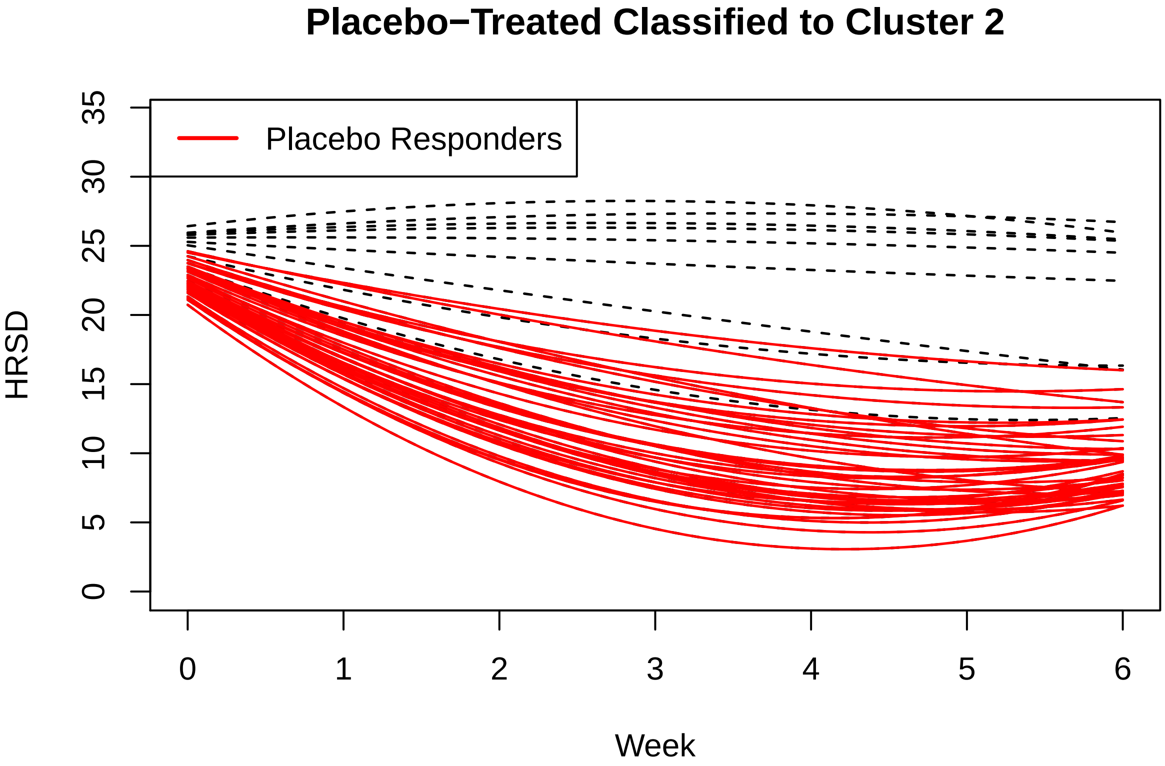
```
# plot trajectories for drug treated subjects in cluster 1
####################FIGURE 7###############################
plot(t, t * 5.7, type = "n", main = "Trajectories for Drug Treated Subjects
     Classified to Cluster 1", xlab = "Week", ylab = "HRSD")
```

```
for (i in 1:prozac$n){
  bi = A %*% as.matrix(prozac$bis[i, ])
  if (drugclass4[i]) lines(tplot, bi[1,1] + bi[2,1] * tplot +
                              bi[3,1] * tplot^2, lwd = 1.25, col = 1)
}
```



```
#####
```

```
# plot trajectories for placebo treated subjects in cluster 2
#####################FIGURE 8################################
indx = (data$cluster == 3)
pboclass3 = indx[1: placebo$n]

plot(t, t*5.7, type="n", main = "Placebo-Treated Classified to Cluster 2",
     xlab = "Week", ylab="HRSD")
for (i in 1:placebo$n){
  bi = A %*% as.matrix(placebo$bis[i, ])
  if (pboclass3[i]) lines(tplot, bi[1, 1] +
                             bi[2, 1] * tplot + bi[3, 1] * tplot^2,
                           lwd = 1.25, col = 1, lty = 2)
}

# Highlight curves of CGI-rated placebo responders
pboclass3res = pboclass3 & (placebo$responder == 1)
for (i in 1:placebo$n){
  bi=A %*% as.matrix(placebo$bis[i, ])
  if (pboclass3res[i]) lines(tplot, bi[1, 1] +
                                bi[2, 1] * tplot + bi[3, 1] * tplot^2,
                              lwd = 1.25, col = 2, lty = 1)
}

legend("topleft", c("Placebo Responders"), col=(2), lwd=c(2), lty=c(1))
```



```
#####
```