Densità miscuglio di componenti Gaussiane

NikolayNikolaev

2023-03-21

Densità miscuglio di componenti Gaussiane

```
funcmxn <- function(x, p, mu, sd){
f1 <- dnorm(x, mu[1], sd[1])
f2 <- dnorm(x, mu[2], sd[2])
f <- p*f1 + (1-p)*f2
f
}
```

```
mu1 <- c(1,4)
sd1 <- c(1,1)
p1 <- 0.4
funcmxn(0.5,
p1,
mu1,
sd1)
```

Scenario 1:

```
## [1] 0.1413497
```

```
y1 <- seq(-5,10,0.01)
length(y1)
```

```
## [1] 1501
```

```
pr1 <- funcmxn(x = y1,
p = p1,
mu = mu1,
sd = sd1)
require(skimr)
```

```
## Loading required package: skimr
```

```
skim_without_charts(pr1)
```

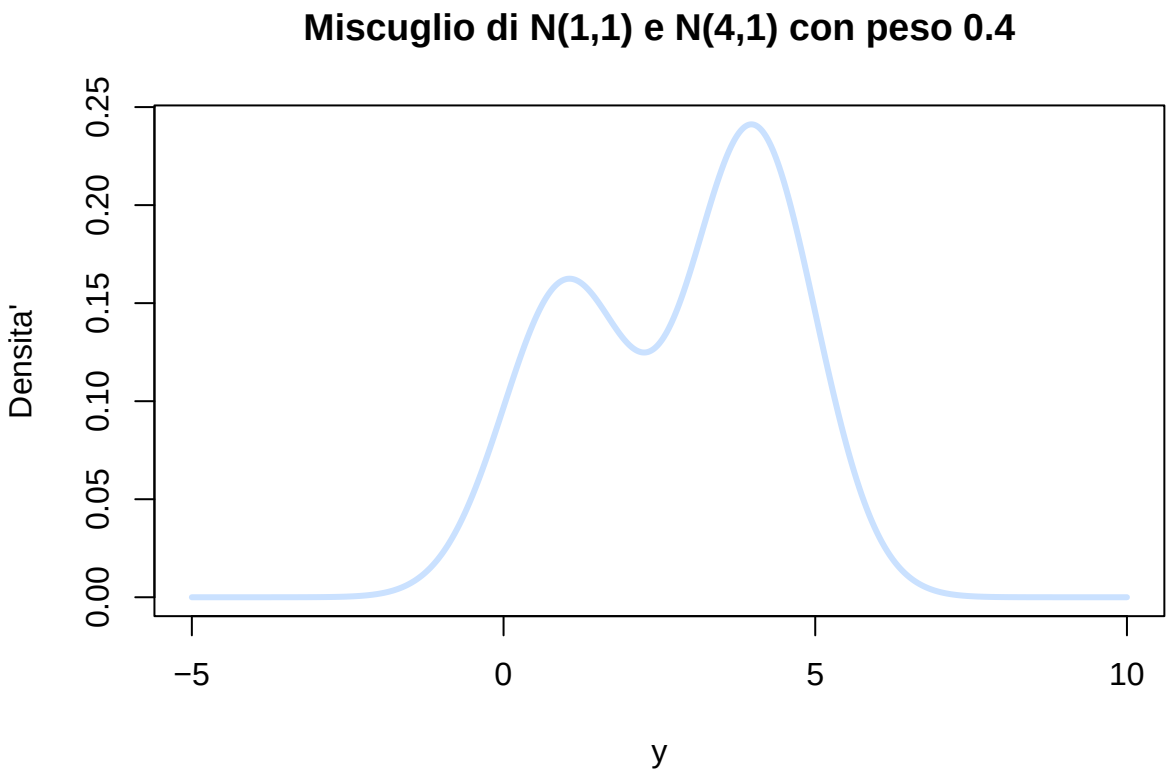
Table 1: Data summary

Name	pr1
Number of rows	1501
Number of columns	1
Column type frequency:	
numeric	1
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
data	0	1	0.07	0.08	0	0	0.02	0.14	0.24

```
plot(y1,
pr1,
xlab = "y",
ylab="Densita'",
lwd=3,
col="lightsteelblue1",
type = "l",
main="Miscuglio di N(1,1) e N(4,1) con peso 0.4")
```



Scenario 2:

```
mu2 <- c(4,4)
sd2 <- c(1,8)
p2 <-0.1
set.seed(1235)
funcmxn(0.5,
p2,
mu2,
sd2)
```

```
## [1] 0.04087215
```

```
y2 <- seq(-30,40,0.01)
pr2 <- funcmxn(x = y2,
p = p2,
mu = mu2,
sd = sd2)
skim_without_charts(pr2)
```

Table 3: Data summary

Name	pr2
Number of rows	7001
Number of columns	1

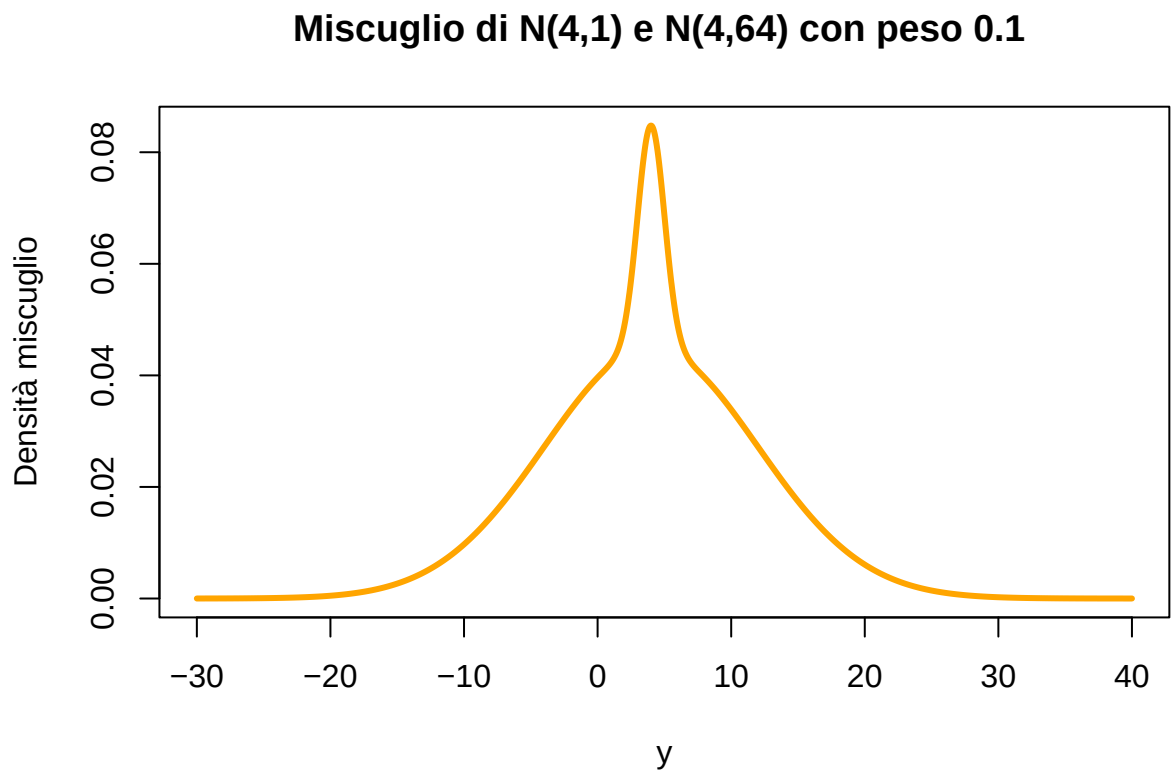
Table 3: Data summary

Column type frequency:	
numeric	1
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
data	0	1	0.01	0.02	0	0	0	0.02	0.08

```
plot(y2,
pr2,
xlab = "y",
ylab="Densità miscuglio",
lwd=3,
col="orange",
type = "l",
main="Miscuglio di N(4,1) e N(4,64) con peso 0.1 ")
```



Scenario 3:

```
mu3 <- c(0,0)
sd3 <- c(1,3)
p3 <-0.5
funcmxn(0.5,
p3,
mu3,
sd3)
```

```
## [1] 0.2416059
```

```
y3 <- seq(-10,20,0.01)
pr3 <- funcmxn(x = y3,
p = p3,
mu = mu3,
sd = sd3)
skim_without_charts(pr3)
```

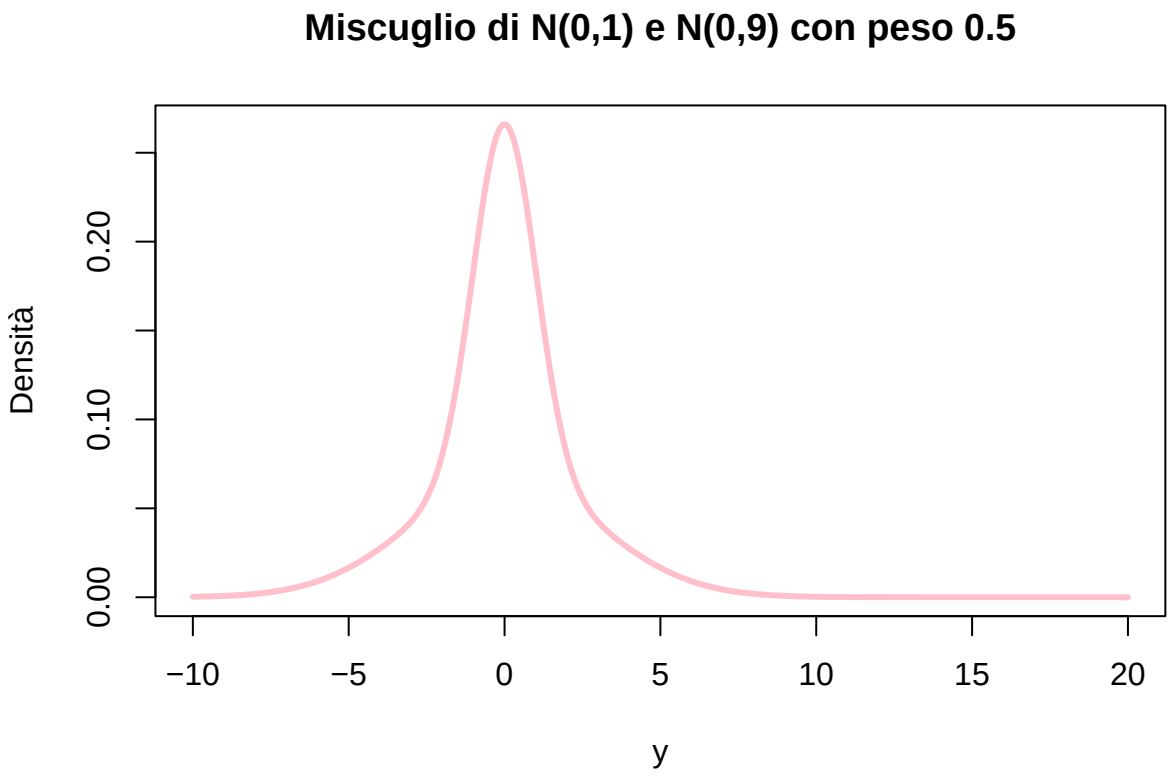
Table 5: Data summary

Name	pr3
Number of rows	3001
Number of columns	1
Column type frequency:	
numeric	1
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
data	0	1	0.03	0.06	0	0	0	0.03	0.27

```
plot(y3,
pr3,
xlab = "y",
ylab="Densità",
lwd=3,
col="Pink",
type = "l",
main="Miscuglio di N(0,1) e N(0,9) con peso 0.5"
)
```



Modello Miscuglio univariato con due componenti Gaussiane

```
load('data/datacol.Rdata')
dim(datacol)
```

```
## [1] 200   3
```

```
head(datacol)
```

```
##     ID cholst sex
## 1 1244    175   1
## 2  835    299   0
## 3  176    250   0
## 4  901    243   0
## 5 1972    150   1
## 6 1994    234   0
```

```
require(skimr)
skim_without_charts(datacol)
```

Table 7: Data summary

Name	datacol
Number of rows	200
Number of columns	3
Column type frequency:	
numeric	3
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
ID	0	1	1344.43	774.26	1	682.5	1367.5	2012.25	2616
cholst	0	1	220.49	43.92	133	191.0	214.0	249.00	357
sex	0	1	0.52	0.50	0	0.0	1.0	1.00	1

```
table(datacol$sex)
```

```
##
##   0   1
##  97 103
```

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## 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
```

```
datacol%>%
dplyr::group_by(sex) %>%
skim_without_charts()
```

Table 9: Data summary

Name	Piped data
Number of rows	200
Number of columns	3

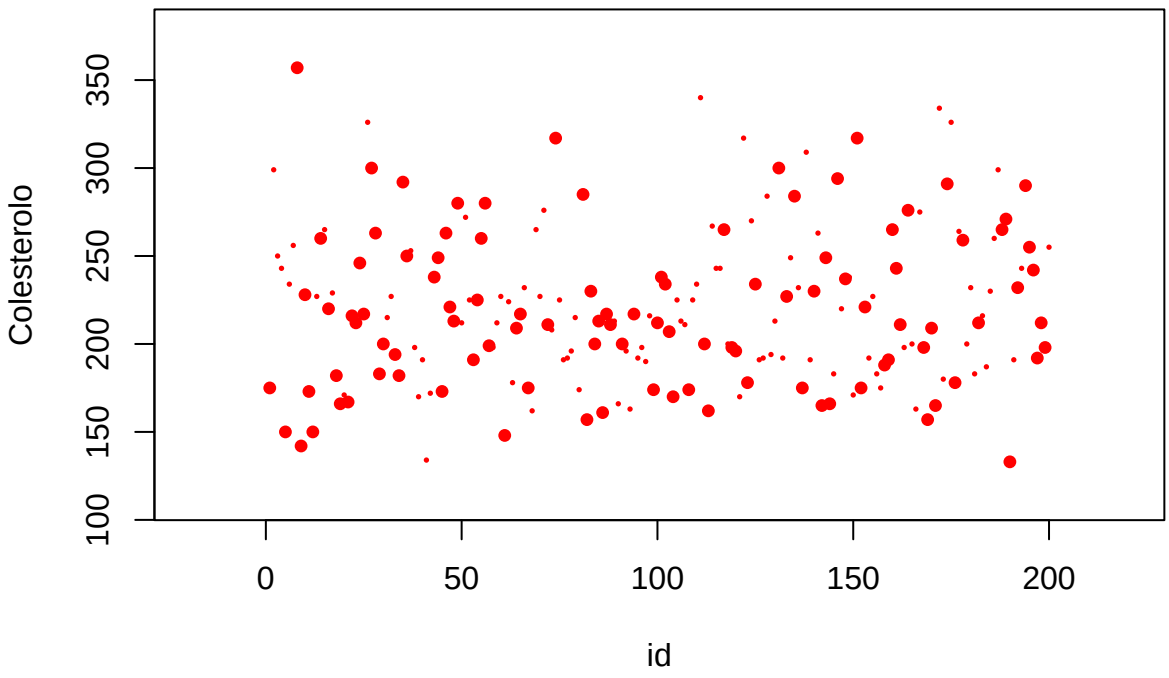
Table 9: Data summary

Column type frequency:	
numeric	2
Group variables	sex

Variable type: numeric

skim_variable	sex	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
ID	0	0	1	1297.75	769.15	2	601	1206	1994	2616
ID	1	0	1	1388.39	780.22	1	831	1442	2069	2597
cholst	0	0	1	222.66	42.67	134	192	216	243	340
cholst	1	0	1	218.45	45.18	133	182	212	249	357

```
n <-dim(datacol)[1]
with(datacol,
symbols(x=1:n,
y=cholst,
circles=sex,
inches=1/30 ,
xlab = "id",
ylab = "Colesterolo",
bg="red",
fg=NULL))
```



Stima dei parametri del modello miscuglio]

```
require('mclust')
```

```
## Loading required package: mclust
```

```
## Package 'mclust' version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.
```

```
mod1 <- Mclust(datacol$cholst,
G = 2,
modelNames = "E")

summary(mod1)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust E (univariate, equal variance) model with 2 components:
##
##  log-likelihood   n df       BIC       ICL
```

```
##       -1032.055 200  4 -2085.302 -2126.187
##
## Clustering table:
##   1   2
## 158  42
```

```
summary(mod1,parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust E (univariate, equal variance) model with 2 components:
##
##  log-likelihood   n df       BIC       ICL
##       -1032.055 200  4 -2085.302 -2126.187
##
## Clustering table:
##   1   2
## 158  42
##
## Mixing probabilities:
##         1         2
## 0.7777503 0.2222497
##
## Means:
##        1        2
## 203.7208 279.1730
##
## Variances:
##        1        2
## 935.6102 935.6102
```

Classificazione delle unita:

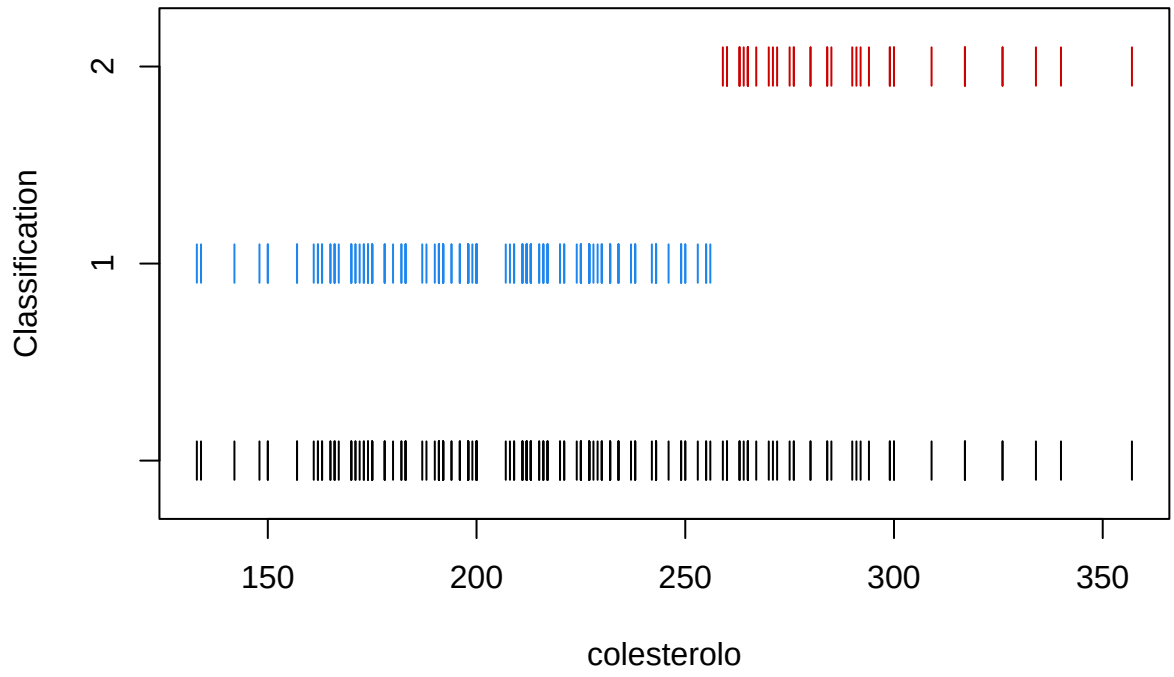
```
head(mod1$z)
```

```
##            [,1]         [,2]
## [1,] 0.99863336 0.0013666414
## [2,] 0.03184899 0.9681510115
## [3,] 0.63194159 0.3680584100
## [4,] 0.75129790 0.2487021035
## [5,] 0.99981810 0.0001819027
## [6,] 0.86199548 0.1380045218
```

```
head(apply(mod1$z,1,which.max))
```

```
## [1] 1 2 1 1 1 1
```

```
plot(mod1, what='classification', xlab = "colesterolo")
```




```
class<-mod1$classification; head(class)
```

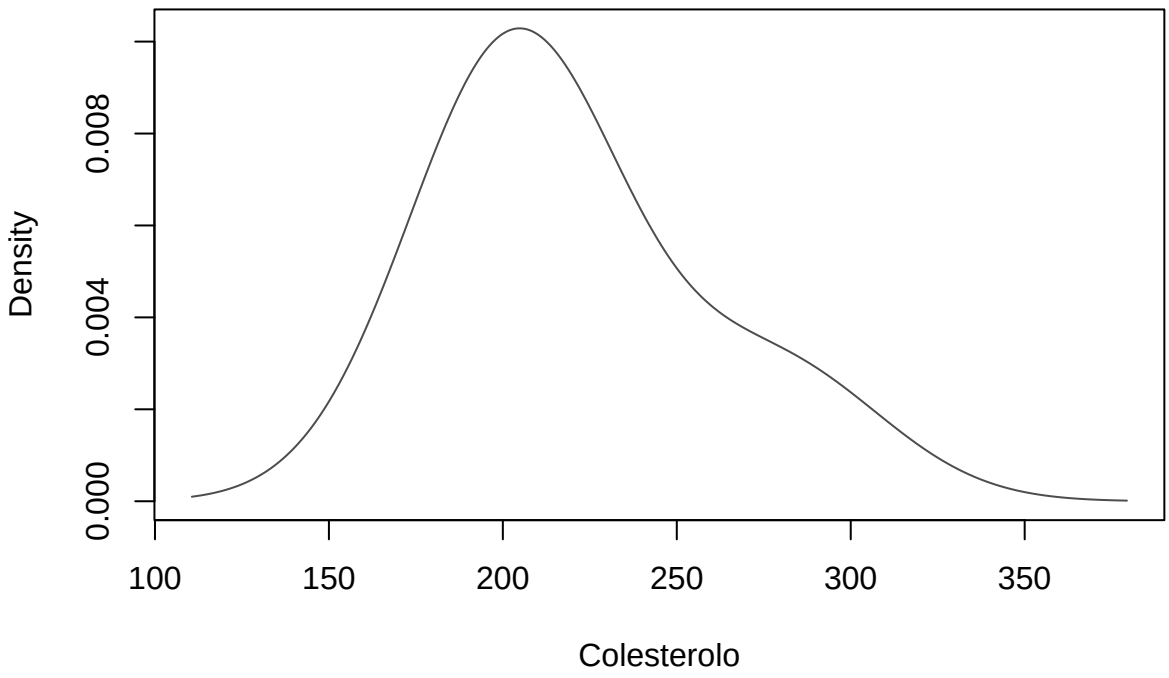
```
## [1] 1 2 1 1 1 1
```

```
#> [1] 1 2 1 1 1 1
table(class,datacol$sex)
```

```
##
## class  0  1
##     1 78 80
##     2 19 23
```

Rapresentazione della densita miscuglio

```
plot(mod1,
what='density', xlab = "Colesterolo")
```

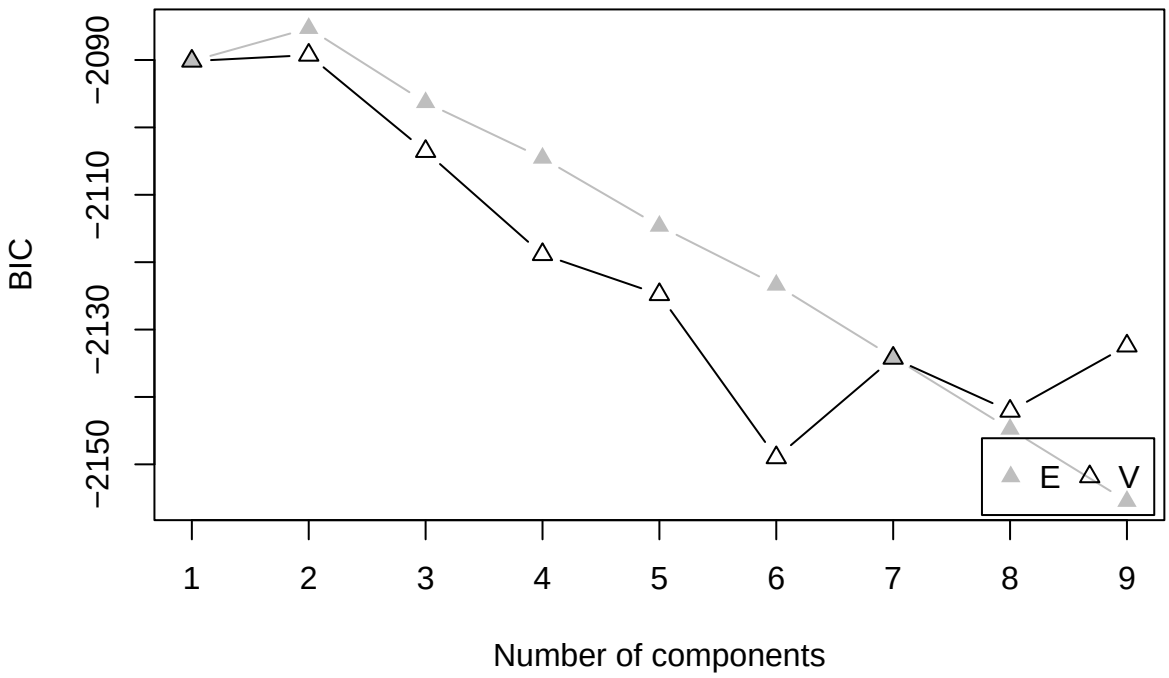


Scelta del numero delle componenti

```
bayesinf <- mclustBIC(datacol$cholst)
bayesinf
```

```
## Bayesian Information Criterion (BIC):
##           E         V
## 1 -2090.155 -2090.155
## 2 -2085.302 -2089.269
## 3 -2096.304 -2103.554
## 4 -2104.550 -2118.857
## 5 -2114.649 -2124.799
## 6 -2123.391 -2149.002
## 7 -2134.099 -2134.237
## 8 -2144.769 -2142.100
## 9 -2155.475 -2132.422
##
## Top 3 models based on the BIC criterion:
##       E,2       V,2       E,1
## -2085.302 -2089.269 -2090.155
```

```
plot(bayesinf)
```



Modello miscuglio MULTIVARIATO con componenti Gaussiane

```
load("data/data.Rdata")
head(data)
```

```
##       Y1.1     Y2.1
## 1 185.2594 222.6894
## 2 209.0801 248.5053
## 4 229.8731 205.3069
## 5 289.8899 218.1793
## 7 254.6754 222.6894
## 8 204.2028 188.8758
```

```
require(skimr)
skim_without_charts(data)
```

Table 11: Data summary

Name	data
Number of rows	200
Number of columns	2

Table 11: Data summary

Column type frequency:	
numeric	2
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Y1.1	0	1	206.39	36.96	100.18	185.26	204.2	229.87	319.39
Y2.1	0	1	212.30	36.11	126.13	184.78	213.8	237.32	302.96

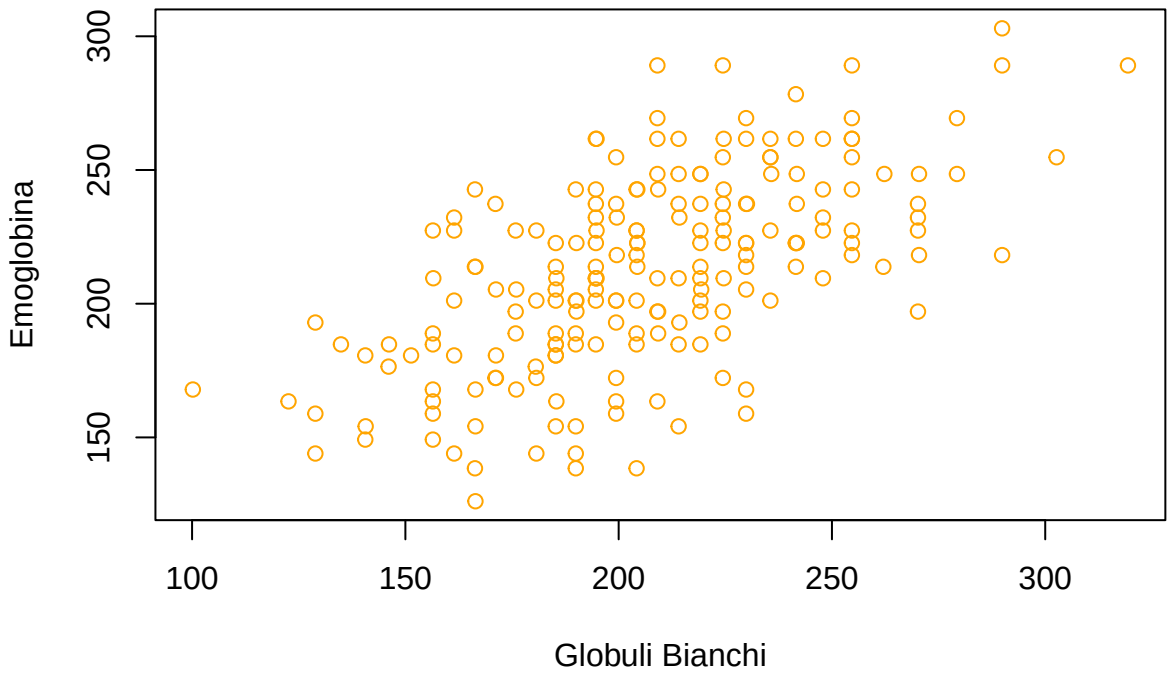
```
cov(data)
```

```
##            Y1.1      Y2.1
## Y1.1 1366.1894  812.4941
## Y2.1  812.4941 1304.0416
```

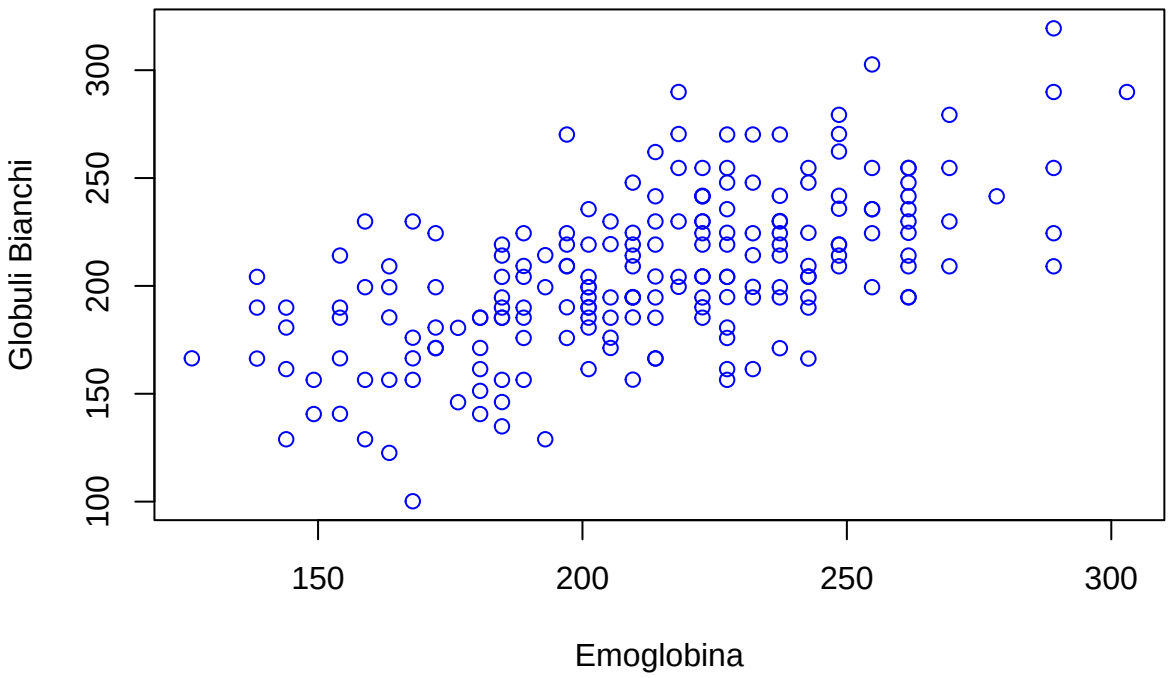
```
cor(data)
```

```
##          Y1.1     Y2.1
## Y1.1 1.000000 0.608722
## Y2.1 0.608722 1.000000
```

```
plot(data$Y1.1, data$Y2.1, xlab = "Globuli Bianchi",
ylab = "Emoglobina", col = "orange")
```




```
plot(data$Y2.1, data$Y1.1,
xlab = "Emoglobina", ylab = "Globuli Bianchi", col = "blue")
```



Selezione del numero delle componenti:

```
require(mclust)
mcc <-Mclust(data, modelNames = c("EII", "VII"))
mcc$BIC
```

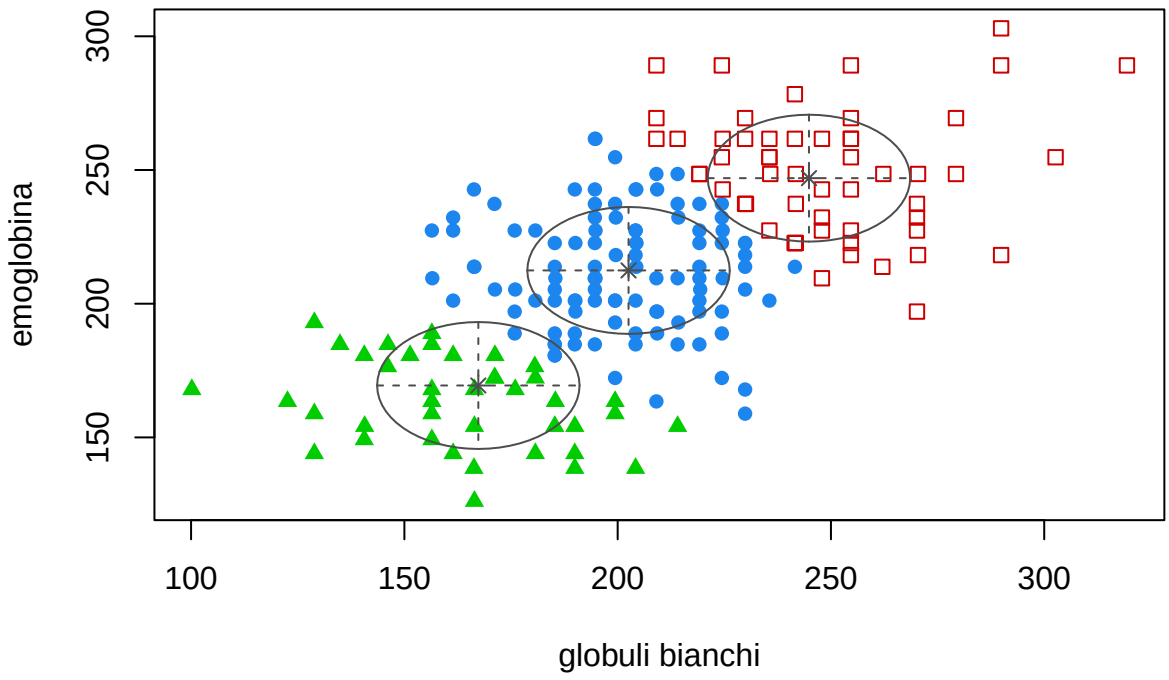
```
## Bayesian Information Criterion (BIC):
##         EII       VII
## 1 -4027.750 -4027.750
## 2 -3972.182 -3977.513
## 3 -3970.082 -3978.606
## 4 -3974.086 -3988.241
## 5 -3989.188 -4007.767
## 6 -4002.328 -4025.495
## 7 -4014.416 -4043.538
## 8 -4030.288 -4058.401
## 9 -4045.601 -4078.349
##
## Top 3 models based on the BIC criterion:
##     EII,3     EII,2     EII,4
## -3970.082 -3972.182 -3974.086
```

Stima dei parametri

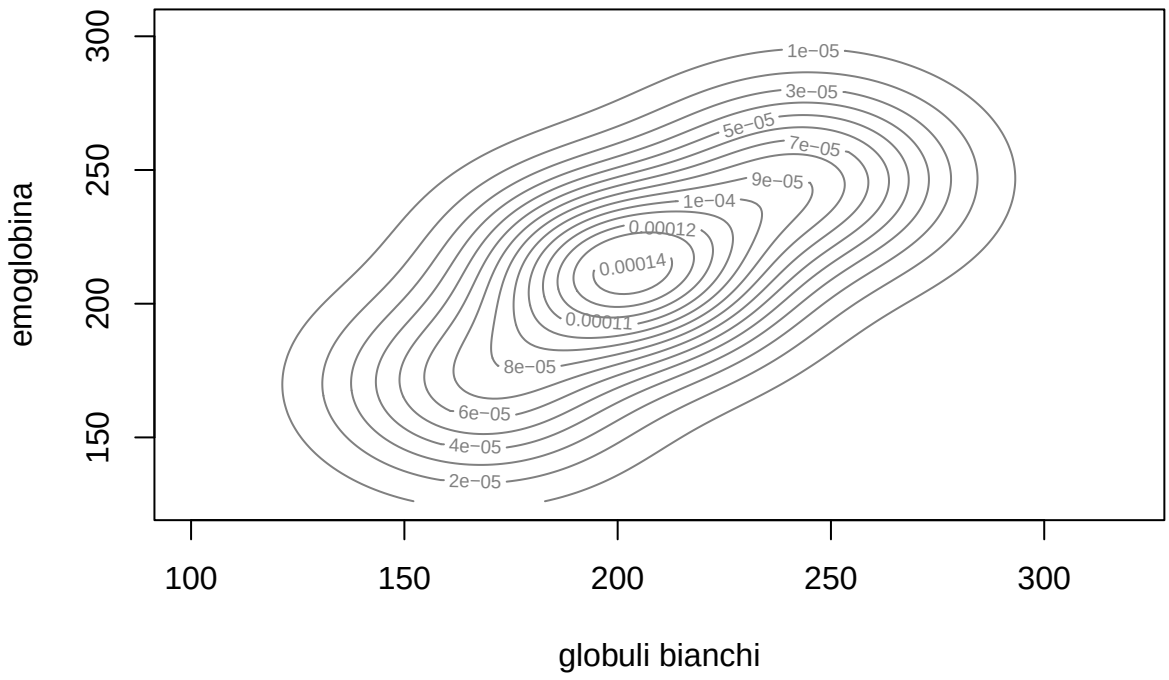
```
mc <-Mclust(data,
G = 3,
modelNames = c("EII"))
summary(mc,parameters = TRUE )
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EII (spherical, equal volume) model with 3 components:
##
##  log-likelihood   n df       BIC       ICL
##       -1961.199 200  9 -3970.082 -4053.728
##
## Clustering table:
##   1   2   3
## 105  55  40
##
## Mixing probabilities:
##         1         2         3
## 0.4897486 0.2813647 0.2288867
##
## Means:
##          [,1]     [,2]     [,3]
## Y1.1 202.5506 244.8475 167.3407
## Y2.1 212.4403 246.9436 169.4106
##
## Variances:
## [,,1]
##          Y1.1     Y2.1
## Y1.1 562.8875   0.0000
## Y2.1   0.0000 562.8875
## [,,2]
##          Y1.1     Y2.1
## Y1.1 562.8875   0.0000
## Y2.1   0.0000 562.8875
## [,,3]
##          Y1.1     Y2.1
## Y1.1 562.8875   0.0000
## Y2.1   0.0000 562.8875
```

```
plot(mc,"classification", xlab = "globuli bianchi",
ylab = "emoglobina")
```




```
plot(mc,"density", xlab = "globuli bianchi", ylab = "emoglobina")
```



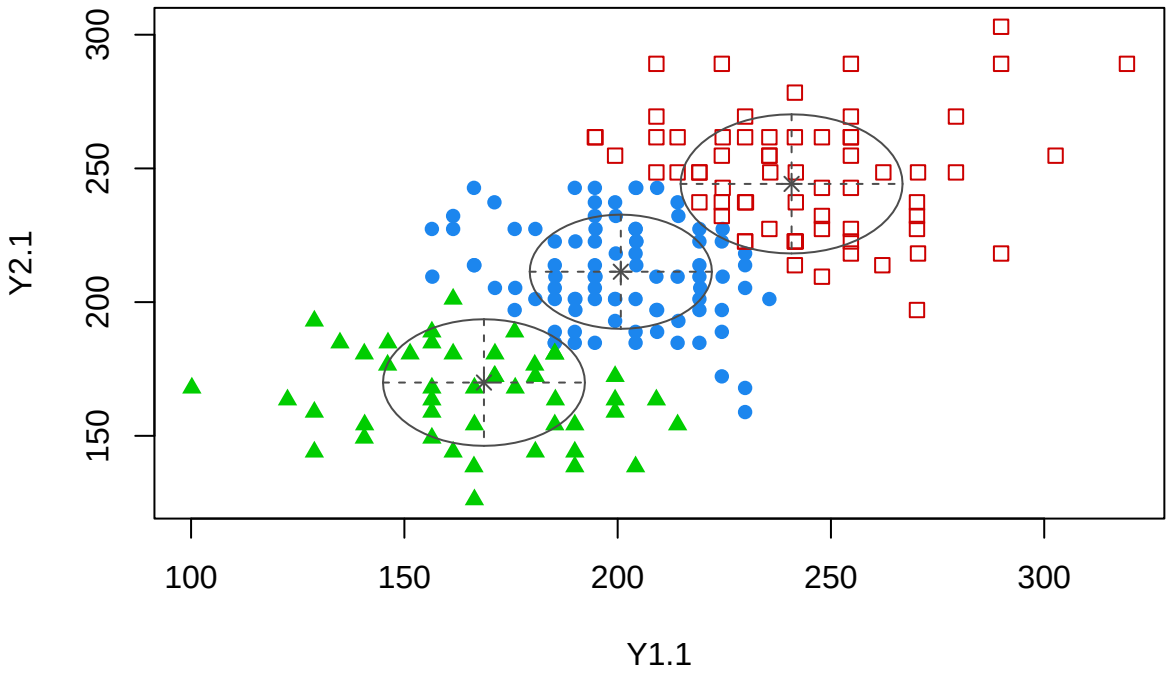
Modello miscuglio con componenti sferiche e varianze specifiche per ogni componente

```
mc1 <-Mclust(data, G = 3, modelNames = c( "VII"))
summary(mc1, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VII (spherical, varying volume) model with 3 components:
##
##  log-likelihood   n df       BIC       ICL
##       -1960.162 200 11 -3978.606 -4064.025
##
## Clustering table:
##  1  2  3
## 88 66 46
##
## Mixing probabilities:
##         1         2         3
## 0.4174926 0.3375221 0.2449853
##
## Means:
##          [,1]     [,2]     [,3]
```

```
## Y1.1 200.7391 240.7749 168.6567
## Y2.1 211.3657 244.2084 169.9287
##
## Variances:
## [,,1]
##          Y1.1     Y2.1
## Y1.1 455.8909   0.0000
## Y2.1   0.0000 455.8909
## [,,2]
##          Y1.1     Y2.1
## Y1.1 676.6679   0.0000
## Y2.1   0.0000 676.6679
## [,,3]
##          Y1.1     Y2.1
## Y1.1 560.0236   0.0000
## Y2.1   0.0000 560.0236
```

```
plot(mc1,"classification")
```

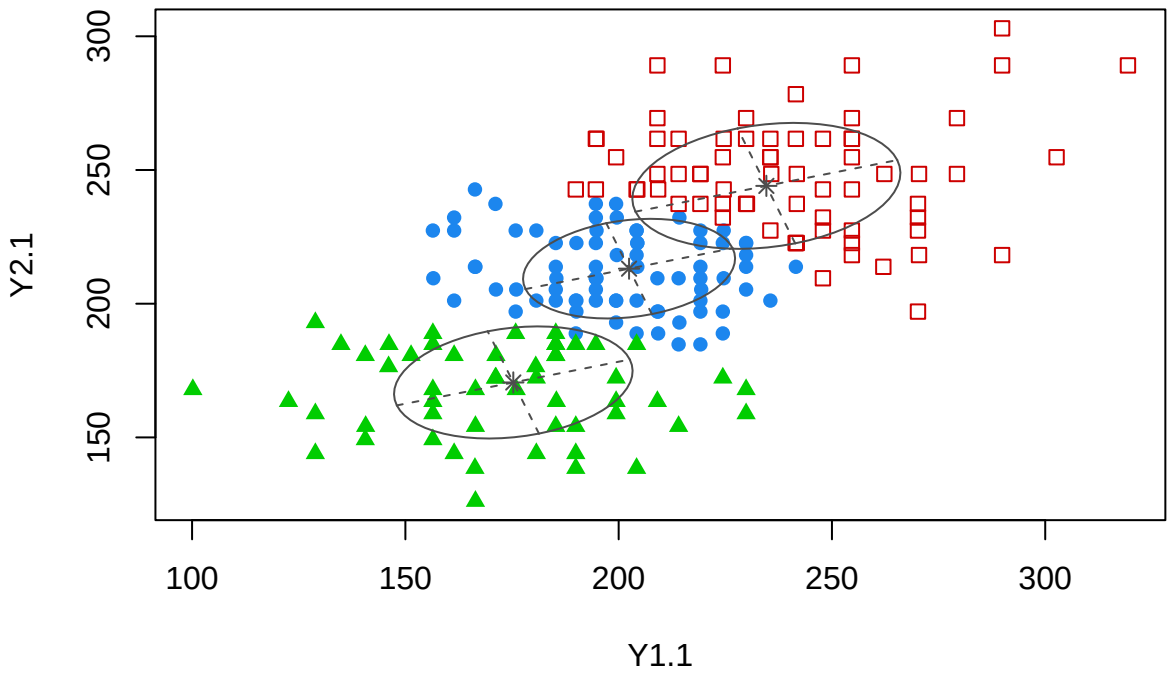


Modello Miscuglio non sferico:

```
mc2 <-Mclust(data, G = 3, modelNames = c( "VEE"))
summary(mc2, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEE (ellipsoidal, equal shape and orientation) model with 3 components:
##
##  log-likelihood   n df       BIC       ICL
##       -1956.648 200 13 -3982.174 -4089.575
##
## Clustering table:
##  1  2  3
## 77 69 54
##
## Mixing probabilities:
##         1         2         3
## 0.3615714 0.3586530 0.2797756
##
## Means:
##          [,1]     [,2]     [,3]
## Y1.1 202.4390 234.6283 175.3050
## Y2.1 213.1036 244.0587 170.5467
##
## Variances:
## [,,1]
##           Y1.1      Y2.1
## Y1.1 617.83969  94.39148
## Y2.1  94.39148 347.15298
## [,,2]
##          Y1.1     Y2.1
## Y1.1 987.4120 150.8535
## Y2.1 150.8535 554.8090
## [,,3]
##          Y1.1     Y2.1
## Y1.1 782.0067 119.4724
## Y2.1 119.4724 439.3955
```

```
plot(mc2,"classification")
```



Bootstrap per errori standard e intervallo di confidenza

```
bootClust <- MclustBootstrap(mc)

summary(bootClust, what = "se")
```

```
## ---------------------------------------------------------
## Resampling standard errors
## ---------------------------------------------------------
## Model                      = EII
## Num. of mixture components = 3
## Replications               = 999
## Type                       = nonparametric bootstrap
##
## Mixing probabilities:
##          1          2          3
## 0.06830526 0.09941863 0.07588911
##
## Means:
##             1        2        3
## Y1.1 7.563568 16.60598 7.475757
## Y2.1 8.214820 11.32911 5.311737
##
## Variances:
```

```
## [,,1]
##         Y1.1    Y2.1
## Y1.1 48.2646  0.0000
## Y2.1  0.0000 48.2646
## [,,2]
##         Y1.1    Y2.1
## Y1.1 48.2646  0.0000
## Y2.1  0.0000 48.2646
## [,,3]
##         Y1.1    Y2.1
## Y1.1 48.2646  0.0000
## Y2.1  0.0000 48.2646
```

```
summary(bootClust, what = "ci")
```

```
## -------------------------------------------------------
## Resampling confidence intervals
## -------------------------------------------------------
## Model                      = EII
## Num. of mixture components = 3
## Replications               = 999
## Type                       = nonparametric bootstrap
## Confidence level           = 0.95
##
## Mixing probabilities:
##               1          2         3
## 2.5%  0.3655952 0.03388683 0.1262837
## 97.5% 0.6446172 0.39446999 0.4370823
##
## Means:
## [,,1]
##           Y1.1     Y2.1
## 2.5%  195.5367 201.2128
## 97.5% 224.9258 233.4173
## [,,2]
##           Y1.1     Y2.1
## 2.5%  234.1689 239.6560
## 97.5% 297.3753 285.3773
## [,,3]
##           Y1.1     Y2.1
## 2.5%  152.7594 162.7716
## 97.5% 181.2393 184.0687
##
## Variances:
## [,,1]
##           Y1.1     Y2.1
## 2.5%  453.0529 453.0529
## 97.5% 639.8634 639.8634
## [,,2]
##           Y1.1     Y2.1
## 2.5%  453.0529 453.0529
## 97.5% 639.8634 639.8634
## [,,3]
##           Y1.1     Y2.1
```

```
## 2.5%  453.0529 453.0529
## 97.5% 639.8634 639.8634
```

Modello a classsi latenti:

```
load("data/psico.Rdata")
dim(Y)
```

```
## [1] 201   2
```

```
head(Y)
```

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    0    0
## [3,]    0    0
## [4,]    1    1
## [5,]    0    0
## [6,]    0    0
```

```
tail(Y)
```

```
##        [,1] [,2]
## [196,]    1    0
## [197,]    2    1
## [198,]    1    1
## [199,]    0    0
## [200,]    3    3
## [201,]    0    0
```

```
n<-dim(Y)[1]
apply(Y,2,table)/n
```

```
##         [,1]       [,2]
## 0 0.35323383 0.39800995
## 1 0.52736318 0.46268657
## 2 0.07960199 0.09950249
## 3 0.03980100 0.03980100
```

```
require(MultiLCIRT)
```

```
## Loading required package: MultiLCIRT
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: limSolve
```

```
Yout <- aggr_data(Y)
S <- Yout$data_dis;S
```

```
##       [,1] [,2]
##  [1,]    1    1
##  [2,]    0    0
##  [3,]    1    0
##  [4,]    2    2
##  [5,]    2    1
##  [6,]    0    1
##  [7,]    1    2
##  [8,]    0    2
##  [9,]    2    0
## [10,]    3    1
## [11,]    3    3
## [12,]    3    2
## [13,]    2    3
## [14,]    1    3
```

```
yv <- Yout$freq; yv
```

```
##  [1] 77 59 20  6  6  7  8  5  1  3  4  1  3  1
```

```
cbind(S,yv)
```

```
##           yv
##  [1,] 1 1 77
##  [2,] 0 0 59
##  [3,] 1 0 20
##  [4,] 2 2  6
##  [5,] 2 1  6
##  [6,] 0 1  7
##  [7,] 1 2  8
##  [8,] 0 2  5
##  [9,] 2 0  1
## [10,] 3 1  3
## [11,] 3 3  4
## [12,] 3 2  1
## [13,] 2 3  3
## [14,] 1 3  1
```

```
mod2 <- est_multi_poly(S,yv,k=2)
```

```
## *-------------------------------------------------------------------------------*
## Link of type =                  0
```

```
## Discrimination index =          0
## Constraints on the difficulty = 0
## Type of initialization =        0
## *-------------------------------------------------------------------------------*
```

```
summary(mod2)
```

```
##
## Call:
## est_multi_poly(S = S, yv = yv, k = 2)
##
## Log-likelihood:
## [1] -374.62
##
## AIC:
## [1] 775.23
##
## BIC:
## [1] 818.18
##
## Class weights:
## [1] 0.4782 0.5218
##
## Conditional response probabilities:
## , , class = 1
##
##          item
## category      1      2
##        0 0.7387 0.8324
##        1 0.2474 0.1024
##        2 0.0138 0.0652
##        3 0.0000 0.0000
##
## , , class = 2
##
##          item
## category      1      2
##        0 0.0000 0.0000
##        1 0.7839 0.7928
##        2 0.1399 0.1309
##        3 0.0763 0.0763
```

```
mod2$np
```

```
## [1] 13
```

```
mod2 <- est_multi_poly(S,yv,k=2,
output = TRUE)
```

```
## *-------------------------------------------------------------------------------*
## Link of type =                  0
## Discrimination index =          0
```

```
## Constraints on the difficulty = 0
## Type of initialization =        0
## *-------------------------------------------------------------------------------*
```

```
round(mod2$Pp,2)
```

```
##       [,1] [,2]
##  [1,] 0.04 0.96
##  [2,] 1.00 0.00
##  [3,] 1.00 0.00
##  [4,] 0.04 0.96
##  [5,] 0.01 0.99
##  [6,] 1.00 0.00
##  [7,] 0.13 0.87
##  [8,] 1.00 0.00
##  [9,] 1.00 0.00
## [10,] 0.00 1.00
## [11,] 0.00 1.00
## [12,] 0.00 1.00
## [13,] 0.00 1.00
## [14,] 0.00 1.00
```