20 - Clustering Based on Statistical Models

Junvie Pailden

SIUE, F2017, Stat 589

November 15, 2017

Clustering Based on Statistical Models

- We introduce statistical models that indicate how the collection of $(p \times 1)$ measurements $\mathbf{x}_j$, from the N objects, was generated.
- Suppose cluster k has proportion p_k of the objects and measurements are generated by a probability density function $f_k(\mathbf{x})$.
- If there are K clusters, the observation vector for a single object is modeled as arising from the mixing distribution

$$f_{Mix}(\mathbf{x}) = \sum_{k=1}^{K} p_k f_k(\mathbf{x})$$

  where each $p_k \geq 0$ and $\sum_{k=1}^{K} p_k = 1$.
- $f_{Mix}(\mathbf{x})$ is called a mixture of the K distributions $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_K(\mathbf{x})$ because the observation is generated from the component distribution $f_k(\mathbf{x})$ with probability p_k

Normal Mixture Model

- Suppose the k-th component $f_k(\mathbf{x})$ is the $N_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ density function.
- The normal mixture model for one observation $\mathbf{x}$ is

$$f_{Mix}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)$$
$$= \sum_{k=1}^{K} p_k \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right)$$

- Clusters generated by this model are ellipsoidal in shape with the heaviest concentration of observations near the center.

Normal Mixture Model (cont.)

- The likelihood function, given N objects and fixed # of clusters K, is

$$
\begin{aligned}
&L(p_1, \ldots, p_K, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K) \\
&\qquad = \prod_{j=1}^{N} f_{Mix}(\mathbf{x}_j | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K) \\
&\qquad = \prod_{j=1}^{N} \left\{ \sum_{k=1}^{K} p_k \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}_k|^{1/2}} \right. \\
&\qquad\qquad \left. \exp\left(-\frac{1}{2} (\mathbf{x}_j - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_k) \right) \right\}
\end{aligned}
$$

Model Selection and Maximum Likelihood Estimates

1. Obtain the MLE $\hat{p}_1, \ldots, \hat{p}_K, \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1, \ldots, \hat{\boldsymbol{\mu}}_K, \hat{\boldsymbol{\Sigma}}_K$ for a fixed number of clsuters K. Let

$$L_{max} = L(\hat{p}_1, \ldots, \hat{p}_K, \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1, \ldots, \hat{\boldsymbol{\mu}}_K, \hat{\boldsymbol{\Sigma}}_K)$$

2. In order to compare models with different numbers of parameters, we compute and use either the AIC or BIC.

Model Selection and Maximum Likelihood Estimates (cont.)

- Akaike Information criterion (AIC)

$$AIC = 2 \ln L_{max} - 2N \left(K \frac{1}{2}(p+1)(p+2) - 1 \right)$$

- Bayesian information criterion (BIC)

$$BIC = 2 \ln L_{max} - 2 \ln(N) \left(K \frac{1}{2}(p+1)(p+2) - 1 \right)$$

3. Select the number of clusters and covariance structure with the largest AIC or BIC.

Structure for the Covariance Matrices $\boldsymbol{\Sigma}_k$

- There is difficulty with too many parameters in the mixture model so simple structures are assumed for the $\boldsymbol{\Sigma}_k$.
- We use the R software package **mclust** to perform model based clustering.

Table 1: Parameterizations of the covariance matrix Σ_k currently available in mclust for hierarchical clustering (HC) and/or EM for multidimensional data. ('•' indicates availability).

identifier	Model	HC	EM	Distribution	Volume	Shape	Orientation
E		•	•	(univariate)	equal		
V		•	•	(univariate)	variable		
EII	λI	•	•	Spherical	equal	equal	NA
VII	λ_k I	•	•	Spherical	variable	equal	NA
EEI	λA		•	Diagonal	equal	equal	coordinate axes
VEI	$\lambda_k A$		•	Diagonal	variable	equal	coordinate axes
EVI	λA_k		•	Diagonal	equal	variable	coordinate axes
VVI	$\lambda_k A_k$		•	Diagonal	variable	variable	coordinate axes
EEE	$\lambda D A D^T$	•	•	Ellipsoidal	equal	equal	equal
EEV	$\lambda D_k A D_k^T$		•	Ellipsoidal	equal	equal	variable
VEV	$\lambda_k D_k A D_k^T$		•	Ellipsoidal	variable	equal	variable
VVV	$\lambda_k D_k A_k D_k^T$	•	•	Ellipsoidal	variable	variable	variable

Figure 1: Structure for the Covariance Matrices

Model Clustering - Old faithful eruptions data I

```
library(mclust)
fit.old <- Mclust(faithful)
summary(fit.old)
```

```
# ----------------------------------------------------
# Gaussian finite mixture model fitted by EM algorithm
# ----------------------------------------------------
#
# Mclust EEE (ellipsoidal, equal volume, shape and orientat
#
#  log.likelihood   n df   BIC   ICL
#           -1126 272 11 -2314 -2361
#
# Clustering table:
#   1   2   3
```

Model Clustering - Old faithful eruptions data II

```
# 130   97   45
```

- In this case, the best model according to BIC is an equal-covariance model with 3 components or clusters.
- A more detailed summary including the estimated parameters can be obtained with the following code:

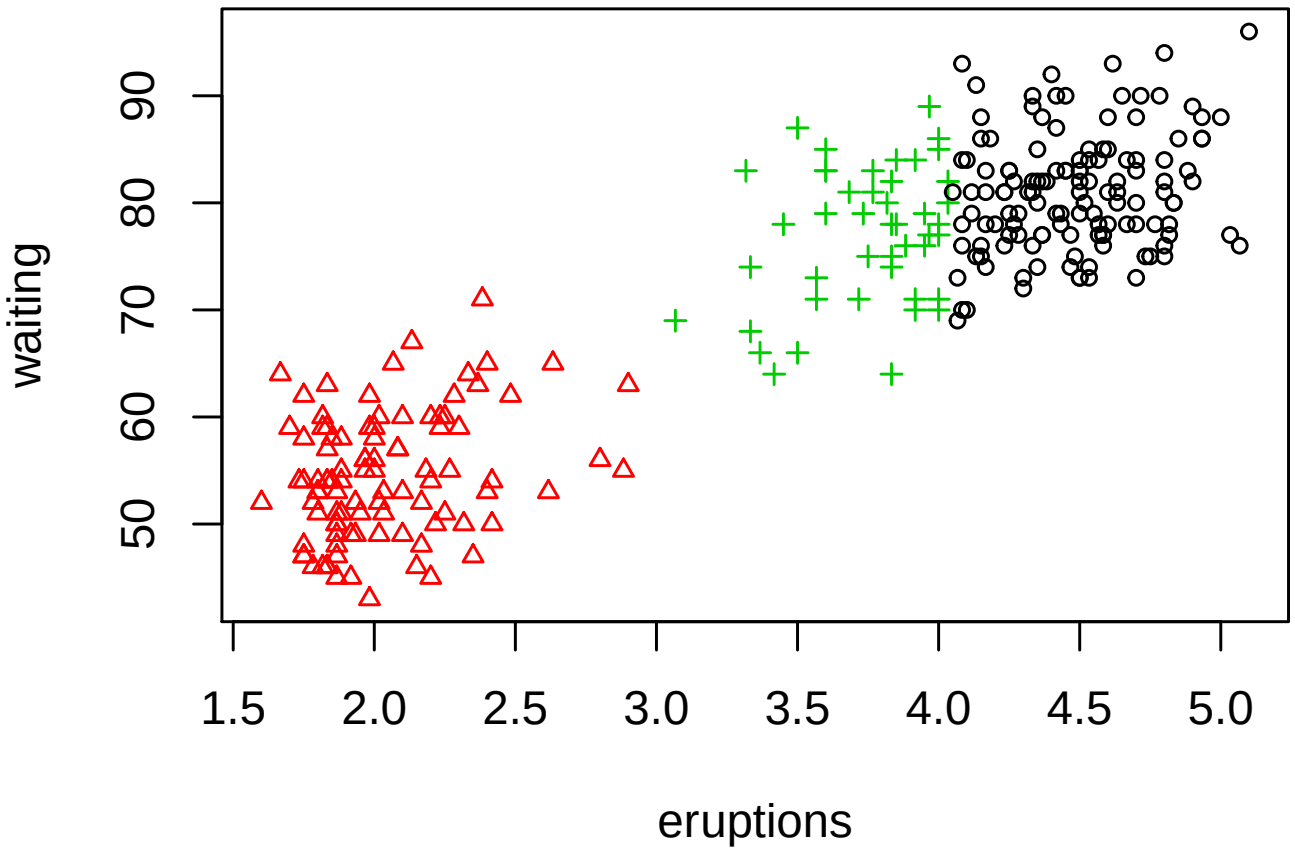
```r
summary(fit.old, parameters = TRUE)
```

```
# ----------------------------------------------------
# Gaussian finite mixture model fitted by EM algorithm
# ----------------------------------------------------
#
# Mclust EEE (ellipsoidal, equal volume, shape and orientat
#
#  log.likelihood   n df   BIC   ICL
#           -1126 272 11 -2314 -2361
#
# Clustering table:
#   1   2   3
# 130  97  45
#
# Mixing probabilities:
```

```
#     1     2     3
# 0.463 0.356 0.180
#
# Means:
#              [,1]  [,2]  [,3]
# eruptions    4.48  2.04  3.82
# waiting     80.89 54.49 77.65
#
# Variances:
# [,,1]
#           eruptions waiting
# eruptions    0.0773   0.476
# waiting      0.4758  33.740
# [,,2]
#           eruptions waiting
# eruptions    0.0773   0.476
# waiting      0.4758  33.740
# [,,3]
```

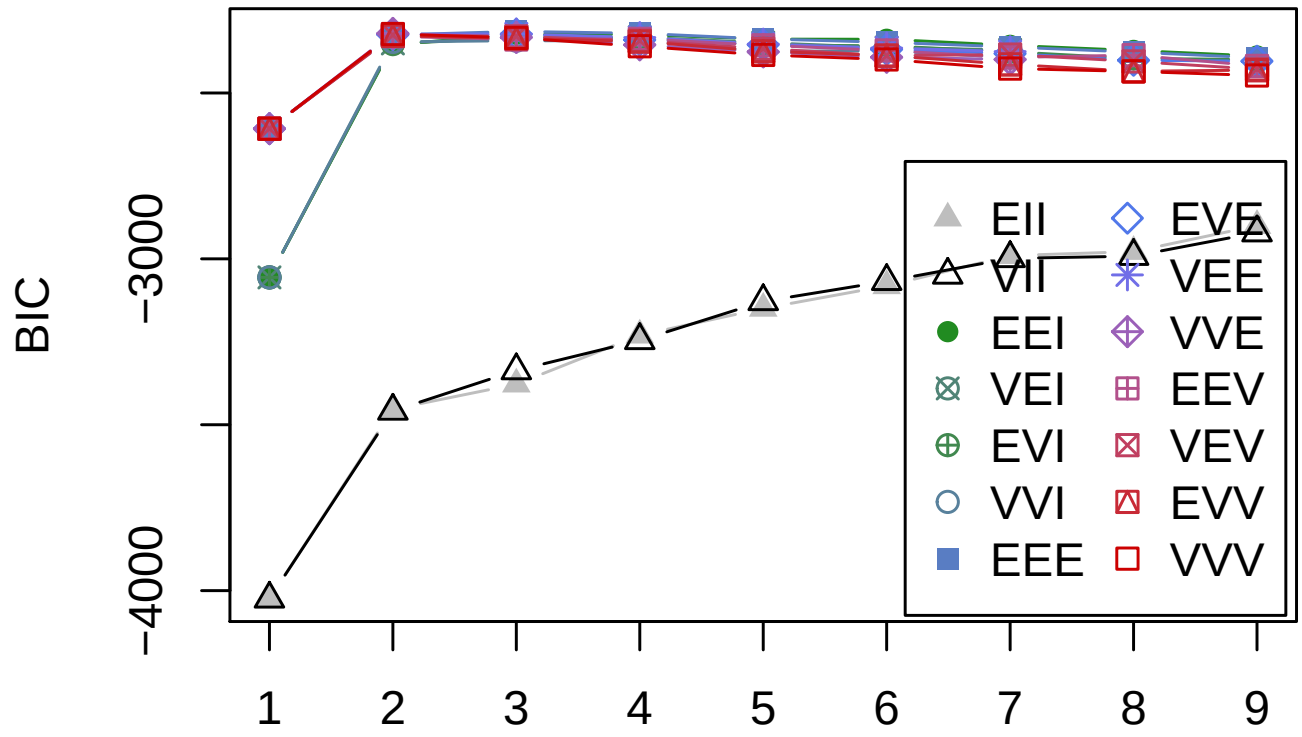
```
#             eruptions waiting
# eruptions      0.0773   0.476
# waiting        0.4758  33.740
```

```
plot(faithful, pch = fit.old$classification,
       col = fit.old$classification,
       cex = 0.7)
```

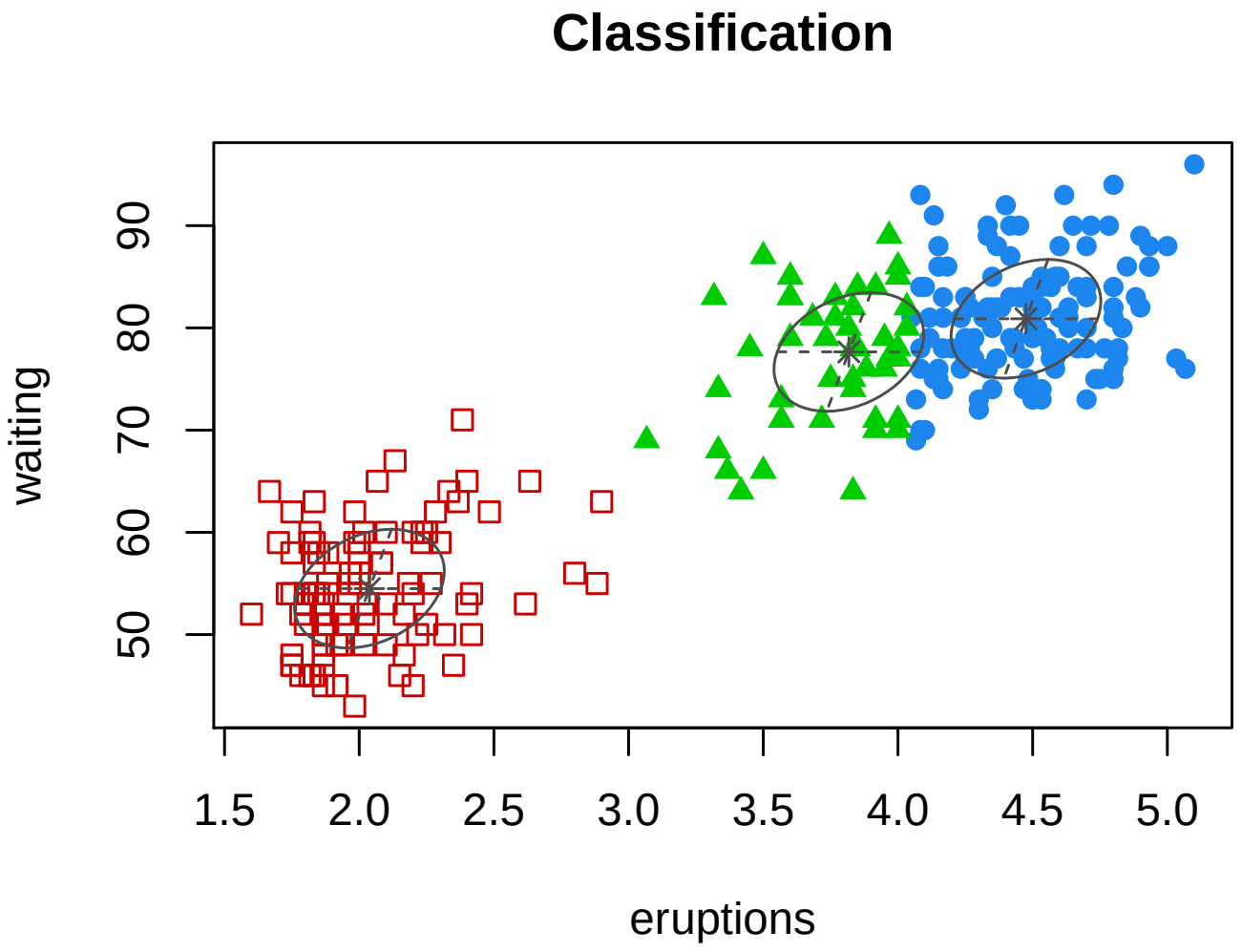

waiting
50
60
70
80
90
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions

Choosing the number of clusters and best covariance structure

```
plot(fit.old, what = "BIC", cex = 0.3)
```



```
plot(fit.old, what = "classification", cex = 0.5)
```



Iris Data I

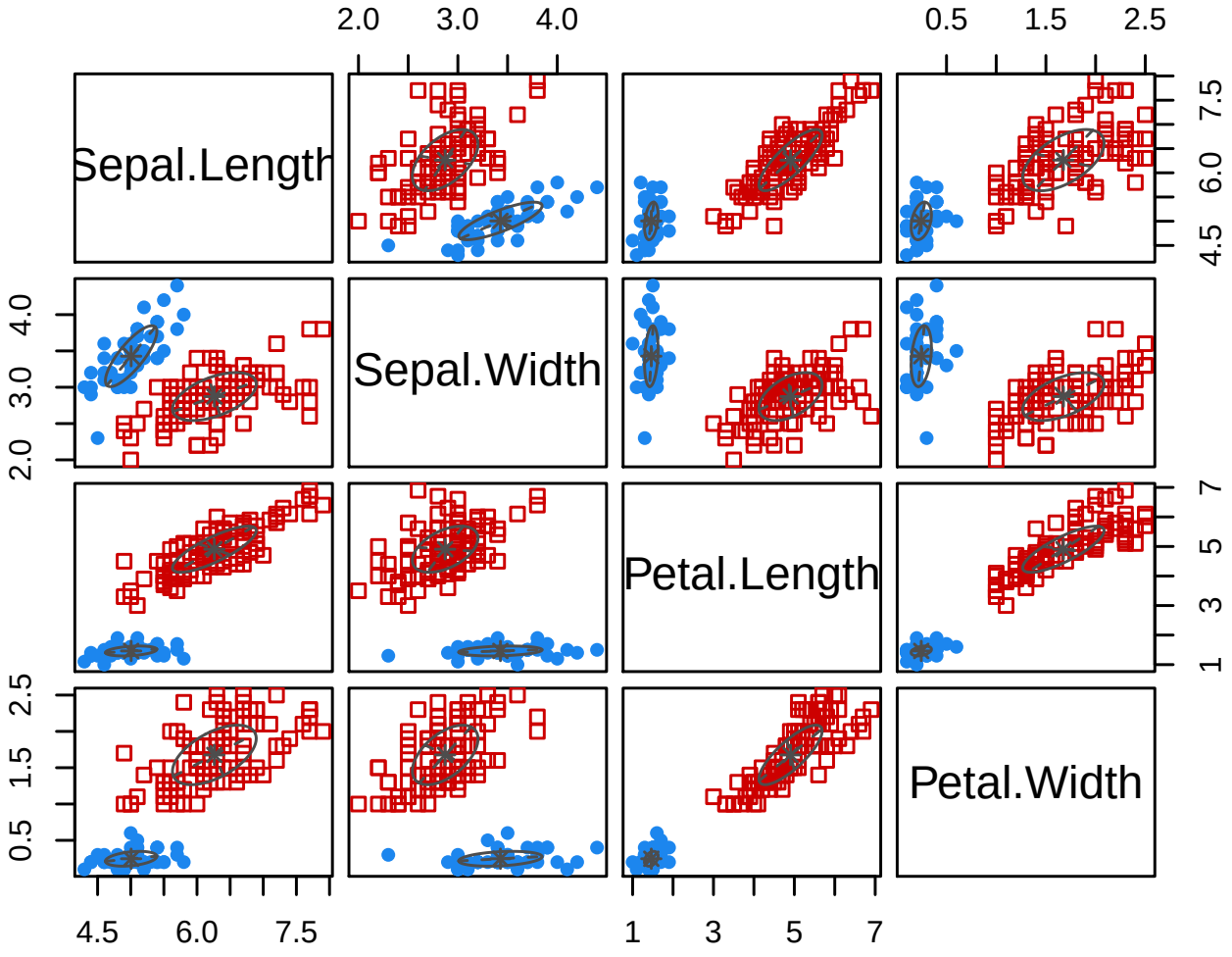
```
fit.iris <- Mclust(iris[,1:4])
summary(fit.iris)
```

```
# ----------------------------------------------------
# Gaussian finite mixture model fitted by EM algorithm
# ----------------------------------------------------
#
# Mclust VEV (ellipsoidal, equal shape) model with 2 compon
#
#  log.likelihood   n df  BIC  ICL
#            -216 150 26 -562 -562
#
# Clustering table:
#   1   2
#  50 100
```

Iris Data II

```
plot(fit.iris, what = "classification", cex = 0.5)
```

Iris Data III



Diabetes Data I

```
data(diabetes)
table(diabetes$class)
```

```
#
# Chemical   Normal    Overt
#       36       76       33
```

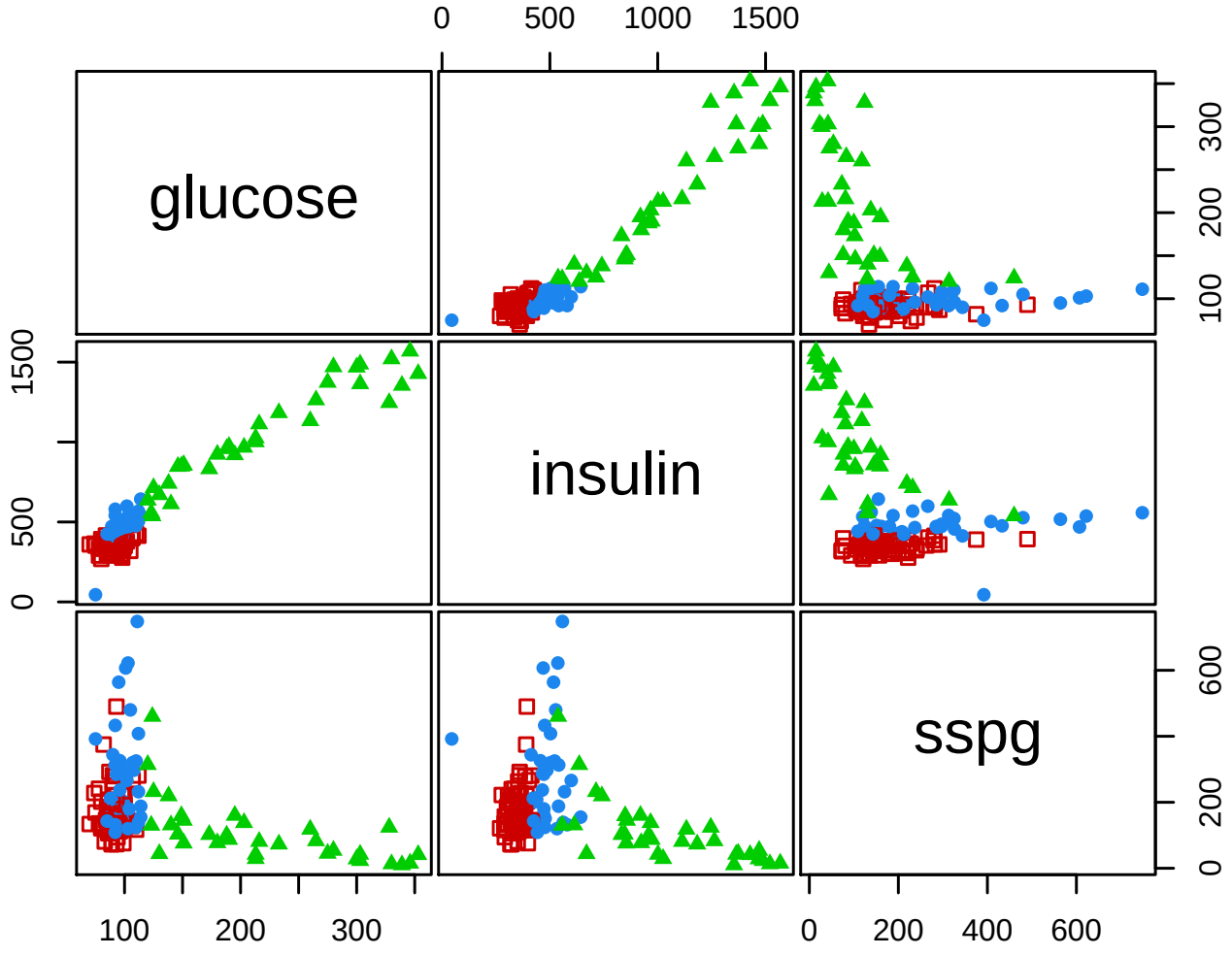
```
X <- diabetes[,-1]
head(X)
```

Diabetes Data II

```
#   glucose insulin sspg
# 1      80     356  124
# 2      97     289  117
# 3     105     319  143
# 4      90     356  199
# 5      90     323  240
# 6      86     381  157
```

```
clPairs(X, diabetes$class)
```

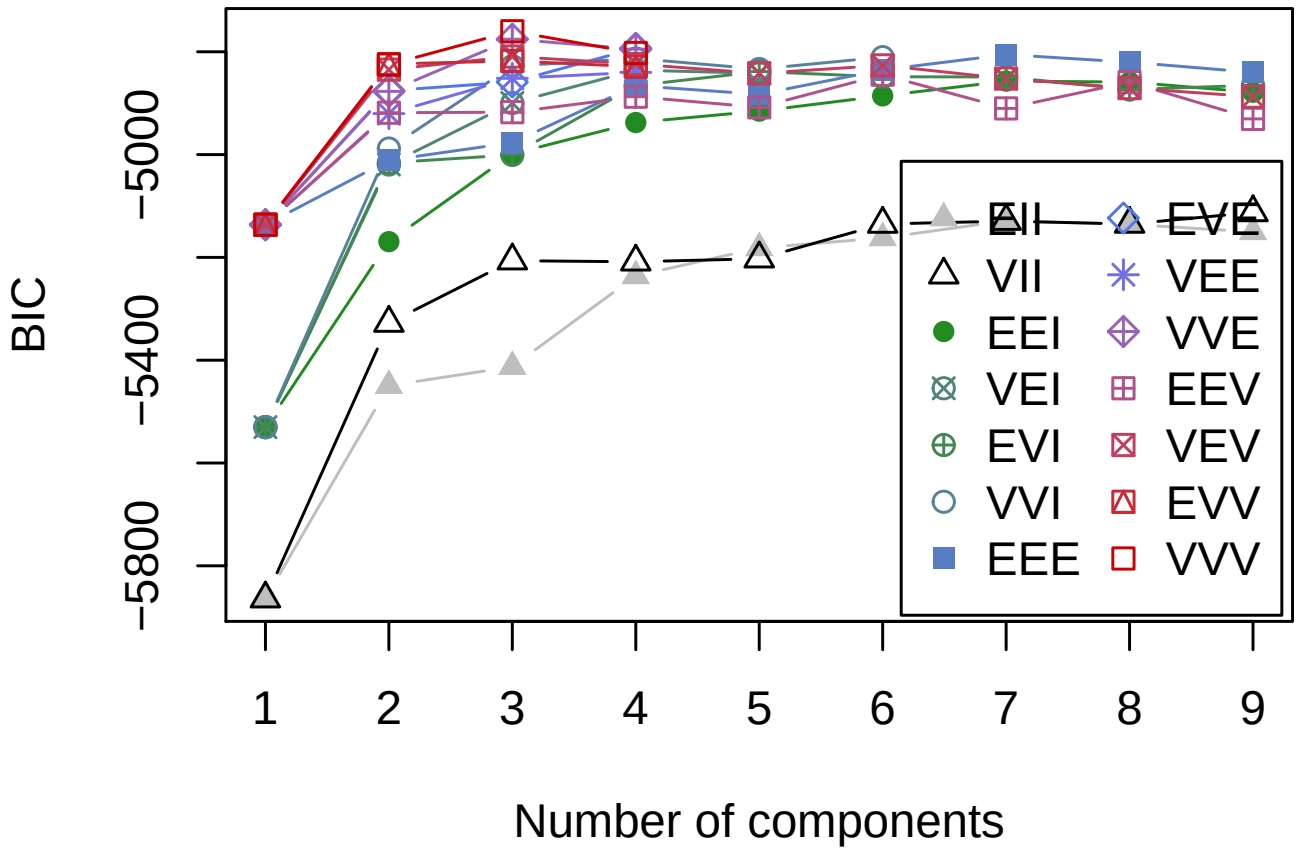
Diabetes Data III



Diabetes Data, # of Clusters I

```
fit.diabetes <- Mclust(X)
plot(fit.diabetes, what  = "BIC", cex = 0.5)
```

Diabetes Data, # of Clusters II



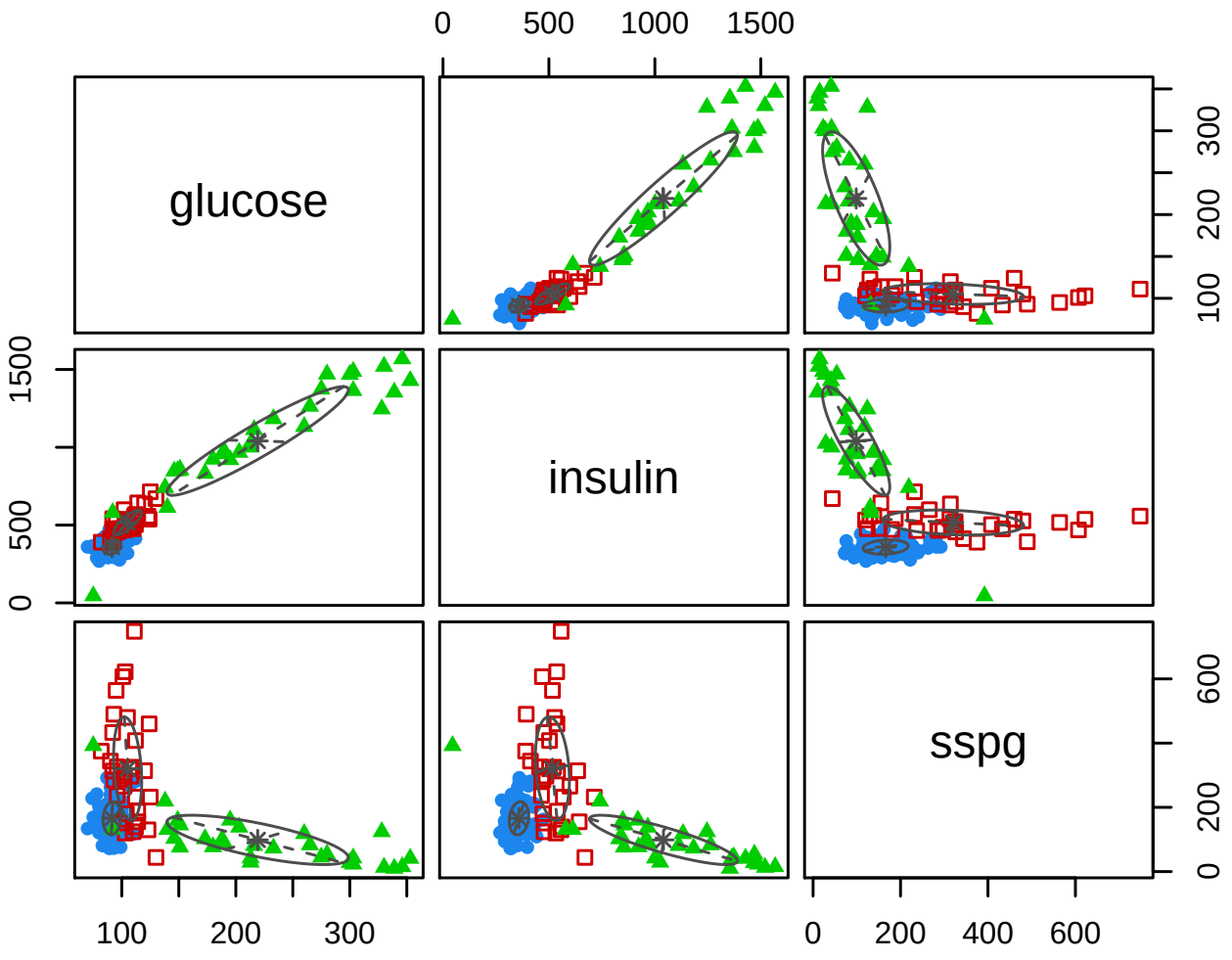
```r
summary(fit.diabetes, parameters = TRUE)
```

```
# ----------------------------------------------------
# Gaussian finite mixture model fitted by EM algorithm
# ----------------------------------------------------
#
# Mclust VVV (ellipsoidal, varying volume, shape, and orien
#
#  log.likelihood   n df   BIC   ICL
#           -2308 145 29 -4760 -4776
#
# Clustering table:
#  1  2  3
# 82 33 30
#
# Mixing probabilities:
```

```
#     1     2     3
# 0.560 0.224 0.215
#
# Means:
#          [,1] [,2]   [,3]
# glucose  91.4  105  219.2
# insulin 358.6  516 1040.6
# sspg    166.0  320   98.6
#
# Variances:
# [,,1]
#         glucose insulin   sspg
# glucose    61.8    97.4   34.4
# insulin    97.4  2107.0  379.0
# sspg       34.4   379.0 2669.1
# [,,2]
#         glucose insulin  sspg
# glucose     152     789  -483
```

```
# insulin     789    6476 -2752
# sspg       -483   -2752 26029
# [,,3]
#         glucose insulin   sspg
# glucose    6351   26190  -4448
# insulin   26190  122126 -22772
# sspg      -4448  -22772   5914
```

```
plot(fit.diabetes, what = "classification", cex = 0.5)
```


glucose
insulin
sspg
0
500
1000
1500
100
200
300
0
500
1500
100
200
300
0
200
400
600
0
200
600