Statistical Learning

Lecture 10a - Unsupervised Learning

ANU - RSFAS

Last Updated: Thu May 12 14:21:11 2022

Unsupervised Learning

- Most of this course focuses on supervised learning methods such as regression and classification.
- In that setting we observe both a set of features $X_1, X_2, \ldots, X_p$ for each object, as well as a response or outcome variable Y. The goal is then to predict Y using $X_1, X_2, \ldots, X_p$.
- Here we instead focus on unsupervised learning, we where observe only the features $X_1, X_2, \ldots, X_p$. We are not interested in prediction, because we do not have an associated response variable Y.

The Goals of Unsupervised Learning

- The goal is to discover interesting things about the measurements: is there an informative way to visualize the data? Can we discover subgroups among the variables or among the observations?

- We will discuss:
  - principal components analysis, a tool used for data visualization or data pre-processing before supervised techniques are applied
  - clustering, a broad class of methods for discovering unknown subgroups in data.

The Challenge of Unsupervised Learning

- Unsupervised learning is more subjective than supervised learning, as there is no simple goal for the analysis, such as prediction of a response.
- But techniques for unsupervised learning are of growing importance in a number of fields:
  - subgroups of breast cancer patients grouped by their gene expression measurements,
  - groups of shoppers characterized by their browsing and purchase histories,
  - movies grouped by the ratings assigned by movie viewers.

Principal Components Analysis

- PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.

- Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization.

First Principle Component

- The first principal component of a set of covariates $X_1, X_2, \ldots, X_P$ is the normalized linear combination of the features:

$$Z_1 = \phi_{11} X_1 + \phi_{12} X_2 + \cdots + \phi_{1p} X_p$$

  that has the largest variance.

- Normalized means that $\sum_{j=1}^{p} \phi_{1j}^2 = 1$

- $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ are called the loadings.

- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.

- The first principal component loading vector solves the optimization problem

$$\underset{\phi_{11},\phi_{12},\ldots,\phi_{1p}}{\text{maximize}} \quad \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{1j}x_{ij}\right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p}\phi_{1j}^2 = 1$$

- This problem can be solved via a singular-value decomposition of the matrix X. This approach also provides the other principle components.

Geometry of PCA

- The loading vector ϕ_1 with elements $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ defines a direction in feature space along which the data vary the most.
- If we project the n data points $x_1, \ldots, x_n$ onto this direction, the projected values are the principal component scores $z_{11}, \ldots, z_{n1}$ themselves.

Further Principal Components

- The second principal component is the linear combination of $X_1, \ldots, X_p$ that has maximal variance among all linear combinations that are uncorrelated with Z_1.

- The second principal component scores $z_{12}, z_{22}, \ldots, z_{n2}$ take the form

$$z_{i2} = \phi_{21} x_{i1} + \phi_{22} x_{i2} + \cdots + \phi_{2p} x_{ip}$$

- It turns out that constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction ϕ_2 to be orthogonal (perpendicular) to the direction ϕ_1. And so on.

Example - Property Loss

- Property losses in USD $100,000 for 58 counties in California
- Data we transformed by $\log_{10}$. Zeroes were kept as zeroes.

```
prop.dat <- read.csv("Cali.csv")
head(prop.dat)
```

```
##      County    Flood  Landsld     Wind Wildfire    Storm  Coastl   Winter
## 1   Alameda 7.245513 6.805976 6.384249 0.000000 4.812913 4.09691 4.462398
## 2    Alpine 6.000000 2.698970 6.137065 8.096910 0.000000 0.00000 5.380350
## 3    Amador 4.018225 0.000000 3.862683 0.000000 0.000000 0.00000 4.779739
## 4     Butte 0.000000 0.000000 5.200821 6.278754 0.000000 0.00000 3.488116
## 5 Calaveras 4.018225 0.000000 3.862683 0.000000 0.000000 0.00000 4.779739
## 6    Colusa 6.146128 0.000000 4.895777 0.000000 0.000000 0.00000 0.000000
```

```
summary(prop.dat)
```

```
##     County              Flood          Landsld          Wind
##  Length:58          Min.   :0.000   Min.   :0.000   Min.   :0.000
##  Class :character   1st Qu.:5.006   1st Qu.:0.000   1st Qu.:4.725
##  Mode  :character   Median :6.334   Median :2.699   Median :5.913
##                     Mean   :5.628   Mean   :2.867   Mean   :5.369
##                     3rd Qu.:7.245   3rd Qu.:6.501   3rd Qu.:6.274
##                     Max.   :8.362   Max.   :7.691   Max.   :7.399
##     Wildfire          Storm           Coastl          Winter
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:3.325
##  Median :0.000   Median :4.301   Median :0.000   Median :4.432
##  Mean   :3.389   Mean   :3.618   Mean   :2.052   Mean   :3.656
##  3rd Qu.:7.307   3rd Qu.:5.799   3rd Qu.:4.234   3rd Qu.:4.780
##  Max.   :9.228   Max.   :8.050   Max.   :7.440   Max.   :6.301
```

```
X <- prop.dat[,-1]
row.names(X) <- prop.dat[,1]
mod.pc <- prcomp(X, center=TRUE, scale=TRUE)
```

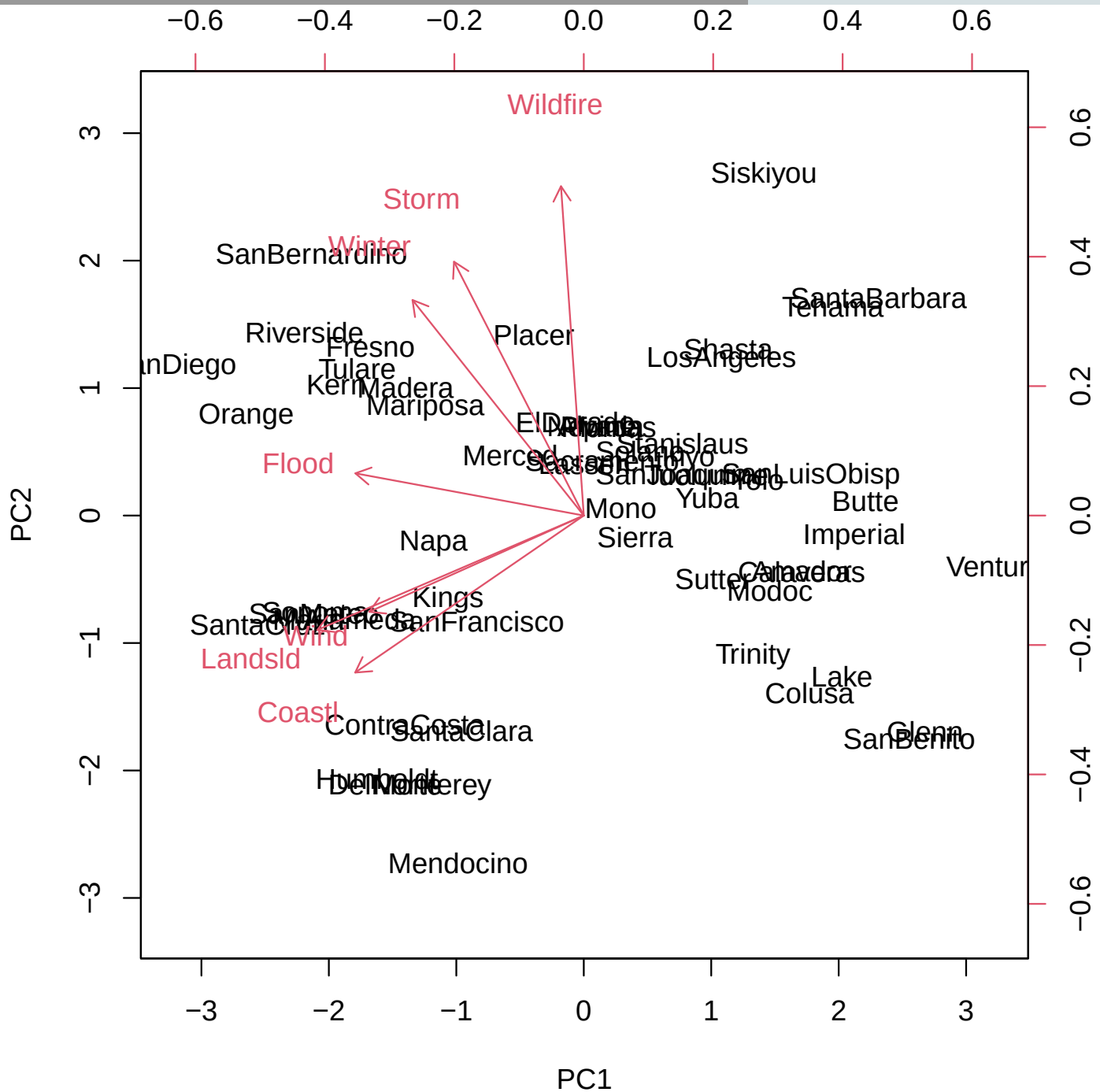
```
summary(mod.pc)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.6305 1.1722 0.9365 0.8828 0.78278 0.68933 0.47231
## Proportion of Variance 0.3798 0.1963 0.1253 0.1113 0.08753 0.06788 0.03187
## Cumulative Proportion  0.3798 0.5761 0.7014 0.8127 0.90025 0.96813 1.00000
```

```
mod.pc
```

```
## Standard deviations (1, .., p=7):
## [1] 1.6304791 1.1722130 0.9365471 0.8828052 0.7827763 0.6893267 0.4723123
##
## Rotation (n x k) = (7 x 7):
##                 PC1         PC2         PC3        PC4        PC5         PC6
## Flood    -0.4411254  0.08159027 -0.44938641  0.2508572  0.2246423  0.65635533
## Landsld  -0.5144452 -0.22086893  0.25609152 -0.1918475  0.1431706  0.13918011
## Wind     -0.4143520 -0.18587887  0.41971438  0.1811698 -0.6867546  0.08397315
## Wildfire -0.0439749  0.63596814  0.55870067 -0.3416839  0.2384152  0.24742517
## Storm    -0.2506104  0.49021210 -0.48295812 -0.4141907 -0.4837622 -0.18030647
## Coastl   -0.4414862 -0.30322420 -0.07738952 -0.4088822  0.3597103 -0.44122145
## Winter   -0.3307833  0.41629400  0.07017627  0.6418381  0.1927355 -0.50435016
##                 PC7
## Flood    -0.22936319
## Landsld   0.73777085
## Wind     -0.32572293
## Wildfire -0.21599862
## Storm     0.15983679
## Coastl   -0.46465399
## Winter    0.09425892
```

```
biplot(mod.pc, scale=0)
```


Wildfire
Storm
Winter
Flood
Landsld
Wind
Coastl
Siskiyou
SanBernardino
SantaBarbara
Riverside
Placer
Fresno
Tulare
Kern
Madera
Mariposa
Orange
Shasta
LosAngeles
Stanislaus
Yuba
Butte
Mono
Sierra
Napa
Imperial
Modoc
Kings
SanFrancisco
Trinity
Lake
Colusa
Glenn
SanBenito
ContraCosta
SantaClara
Mendocino
PC1
PC2

Variance Explained by each PC

```
mod.pc$sdev^2
```

```
## [1] 2.6584621 1.3740833 0.8771206 0.7793451 0.6127388 0.4751713 0.2230789
```

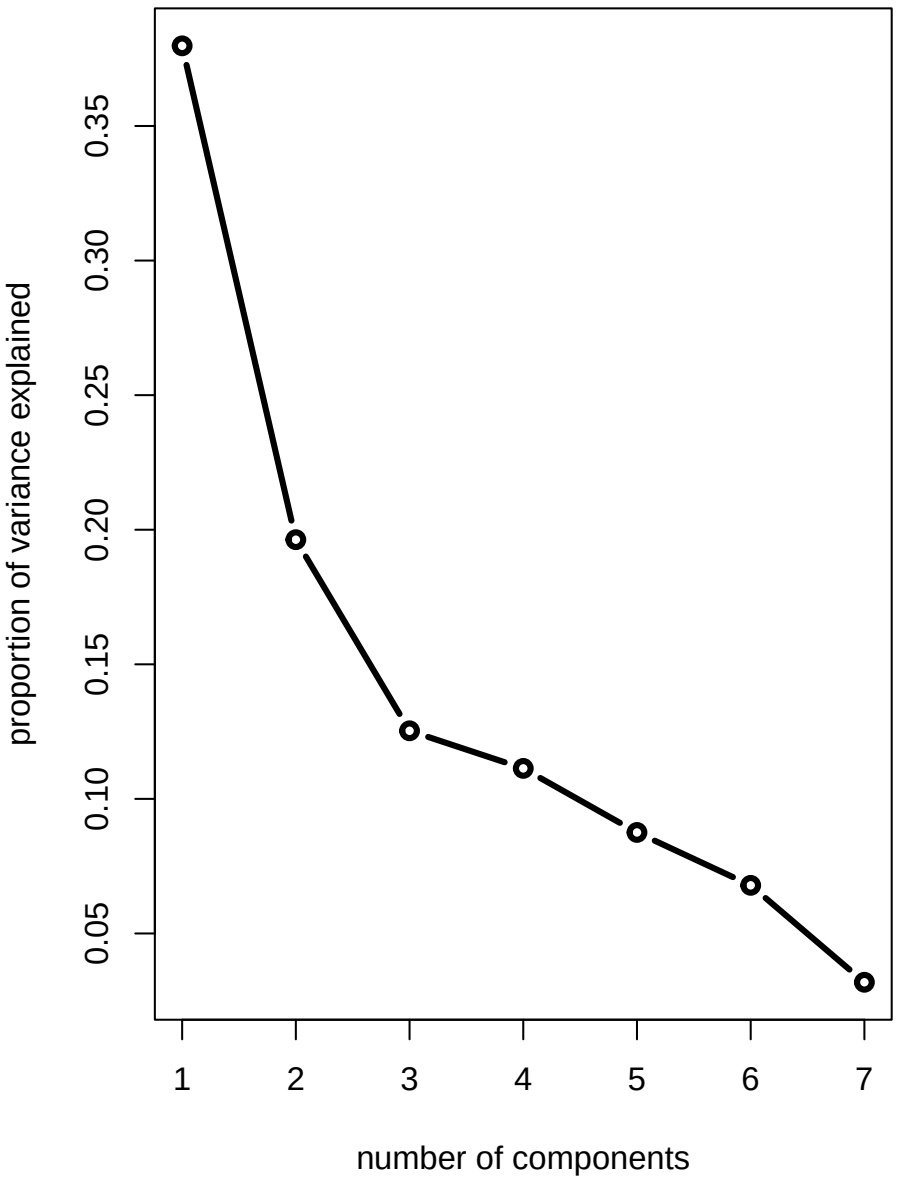
```
mod.pc$sdev^2/sum(mod.pc$sdev^2)
```

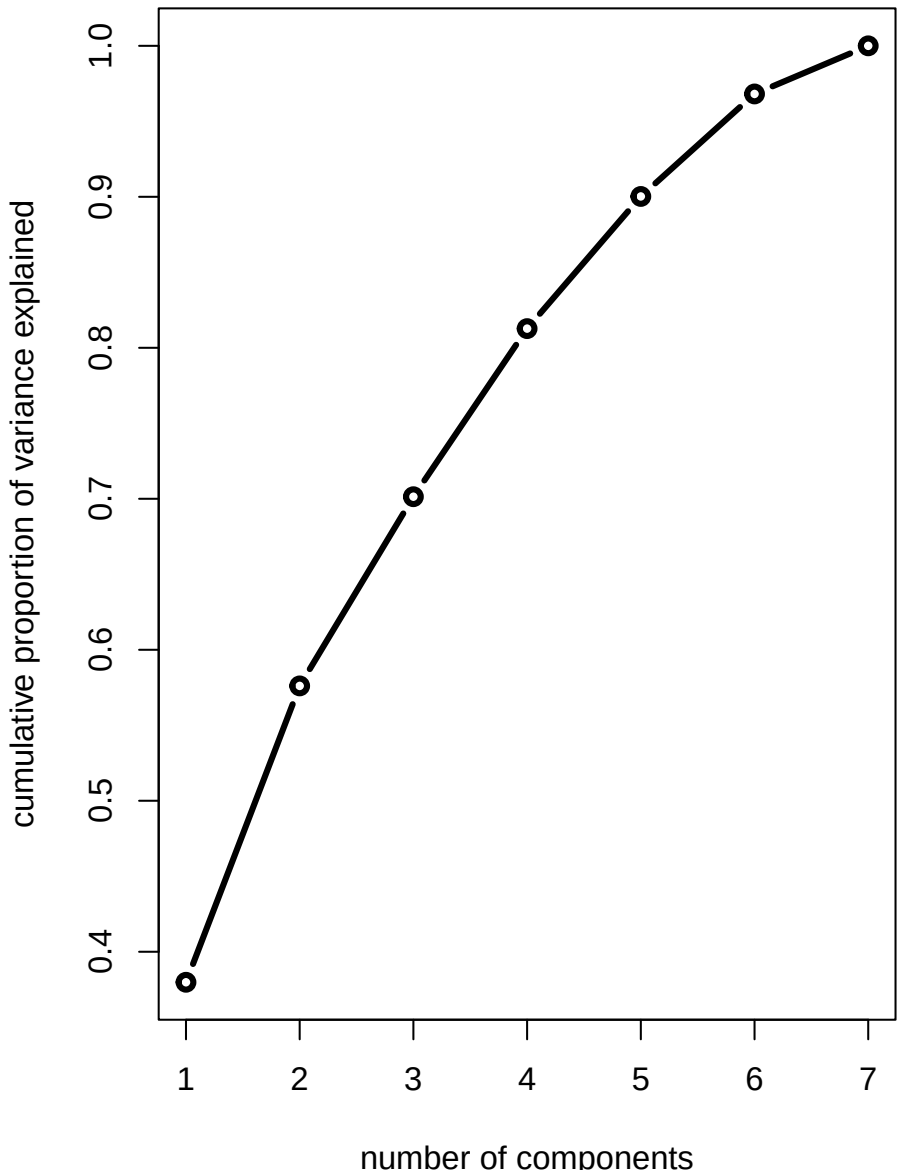
```
## [1] 0.37978029 0.19629761 0.12530294 0.11133501 0.08753412 0.06788161 0.03186842
```

```
cumsum(mod.pc$sdev^2/sum(mod.pc$sdev^2))
```

```
## [1] 0.3797803 0.5760779 0.7013808 0.8127159 0.9002500 0.9681316 1.0000000
```

```
par(mfrow=c(1,2))
plot(mod.pc$sdev^2/sum(mod.pc$sdev^2), type="b", lwd=3,
     xlab="number of components", ylab="proportion of variance explained")
plot(cumsum(mod.pc$sdev^2/sum(mod.pc$sdev^2)), type="b", lwd=3,
     xlab="number of components", ylab="cumulative proportion of variance explained")
```





Clustering

- Clustering refers to a very broad set of techniques for finding subgroups, or clusters, in a data set.
- We seek a partition of the data into distinct groups so that the observations within each group are quite similar to each other,
- To make this concrete, we must define what it means for two or more observations to be similar or different.
- Indeed, this is often a domain-specific consideration that must be made based on knowledge of the data being studied.

PCA vs Clustering

- PCA looks for a low-dimensional representation of the observations that explains a good fraction of the variance.
- Clustering looks for homogeneous subgroups among the observations.

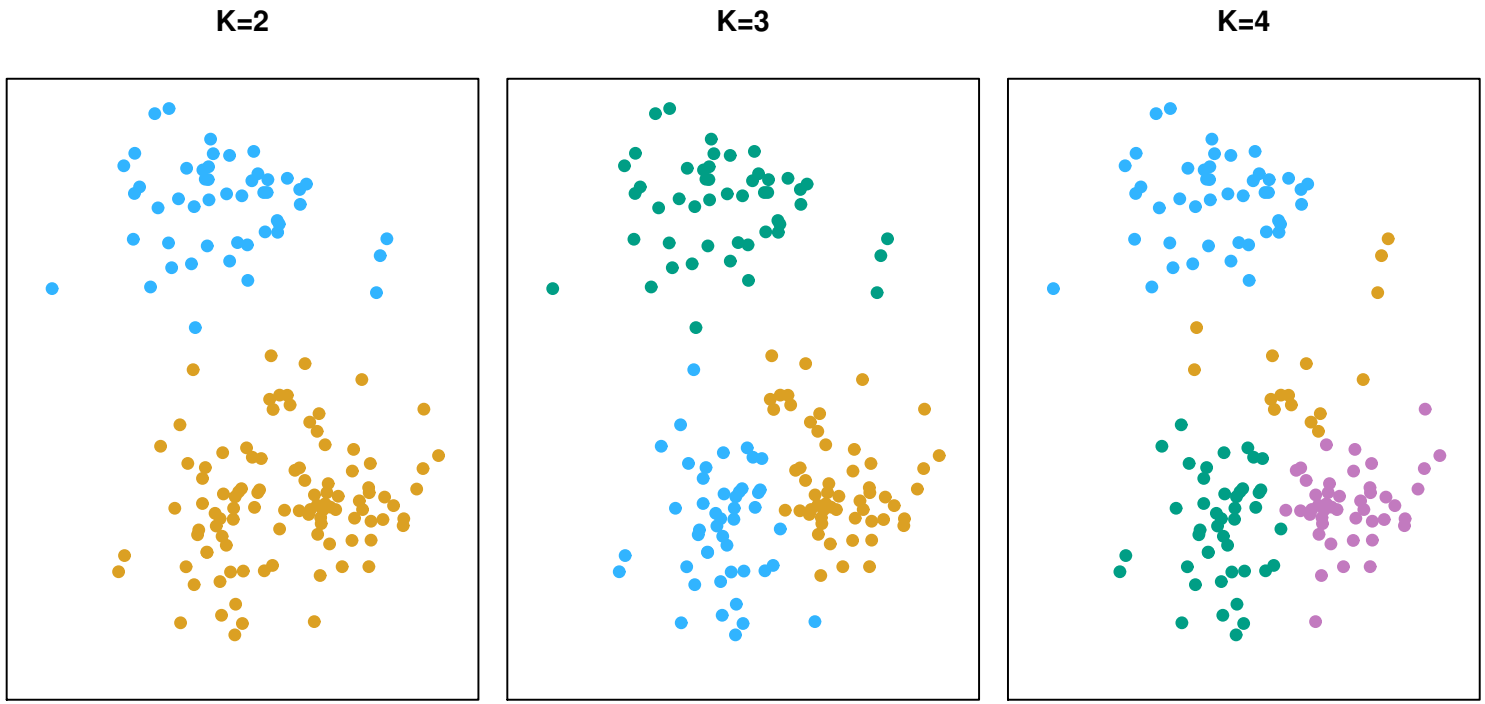
Example: Clustering for Market Segmentation

- Suppose we have access to a large number of measurements (e.g. median household income, occupation, distance from nearest urban area, and so forth) for a large number of people.
- Our goal is to perform market segmentation by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a particular product.
- The task of performing market segmentation amounts to clustering the people in the data set.

Some Clustering Methods

- K-means clustering: we seek to partition the observations into a pre-specified number of clusters.
- Hierarchical clustering: we do not know in advance how many clusters we want; in fact, we end up with a tree-like visual representation of the observations, called a dendrogram, that allows us to view at once the clustering obtained for each possible number of clusters, from 1 to n.
- Model based clustering: we formally write down a mixture of probability distributions and use selection criterion such BIC to determine the number of clusters.

K-means Clustering



- A simulated data set with 150 observations in 2-dimensional space.

Details of K-means Clustering

- Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster. These satisfy two properties:
  1. $C_1 \cup C_2 \cup \ldots \cup C_k = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.
  2. $C_k \cap C_k = 0$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

- The idea behind K-means clustering is that a good clustering is one for which the within-cluster variation is as small as possible.

- The within-cluster variation for cluster C_k is a measure $WCV(C_k)$ of the amount by which the observations within a cluster differ from each other.

- Hence we want to solve the problem:

$$\underset{C_1, C_2, \ldots, C_K}{\text{minimize}} \left\{ \sum_{i=1}^{K} WCV(C_k) \right\}$$

- In words, this formula says that we want to partition the observations into K clusters such that the total within-cluster variation, summed over all K clusters, is as small as possible.

How to Define Within-Cluster Variation?

- Typically we use Euclidean distance

$$WCV(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

where $|C_k|$ denotes the number of observations in the k^{th} cluster.

K-Means Clustering Algorithm

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:

   2.1 For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   2.2 Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

Properties of the Algorithm

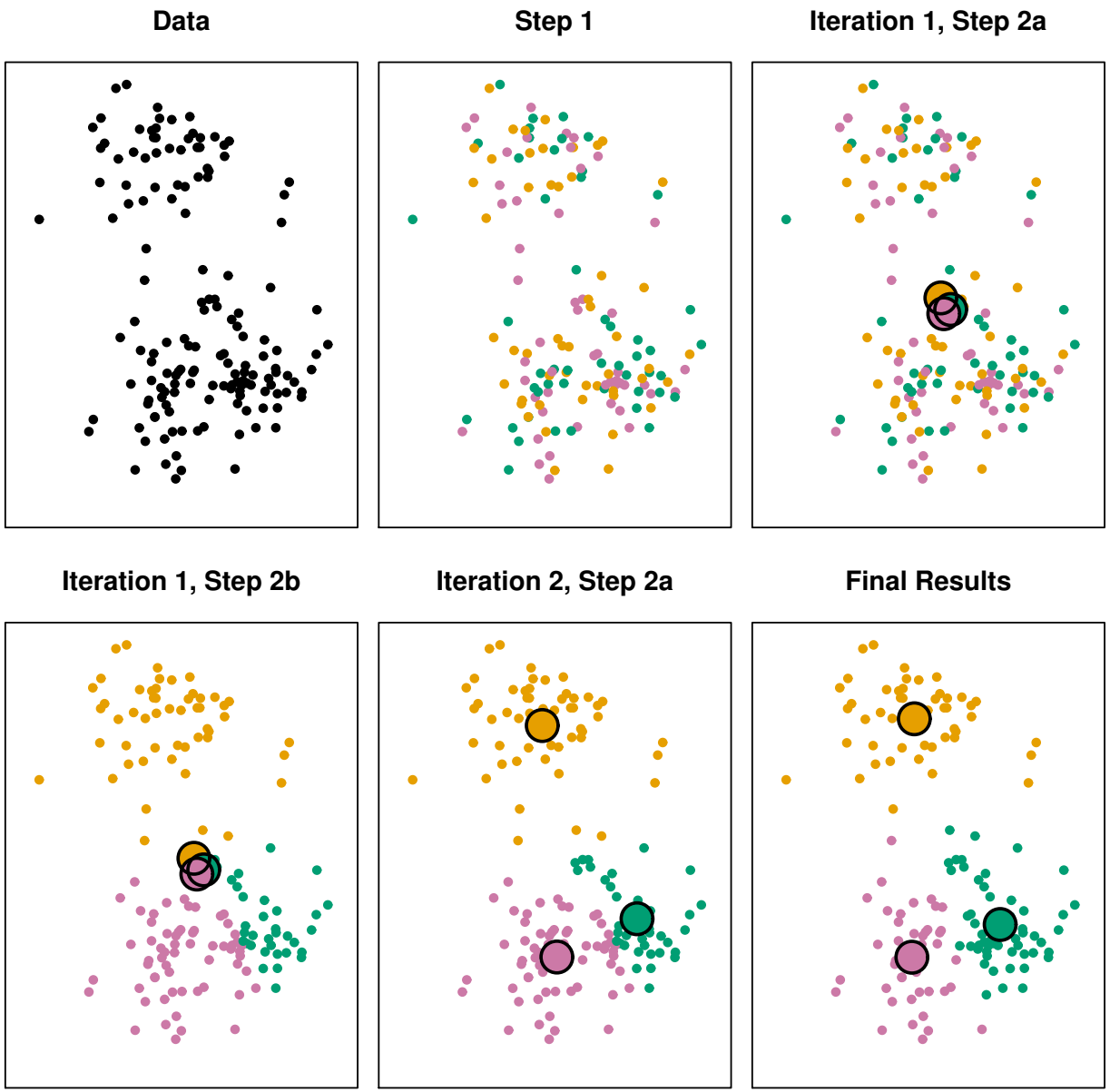
- This algorithm is guaranteed to decrease the value of the objective function at each step.
- Note we can rewrite the WCV as:

$$\frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i \in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2$$

Where $\bar{x}_{kj} = \frac{1}{|C_k|\sum_{i \in C_k} x_{ij}}$ is the mean for feature j in cluster C_k.

- However, the algorithm is not guaranteed to find the global minimum.

Example

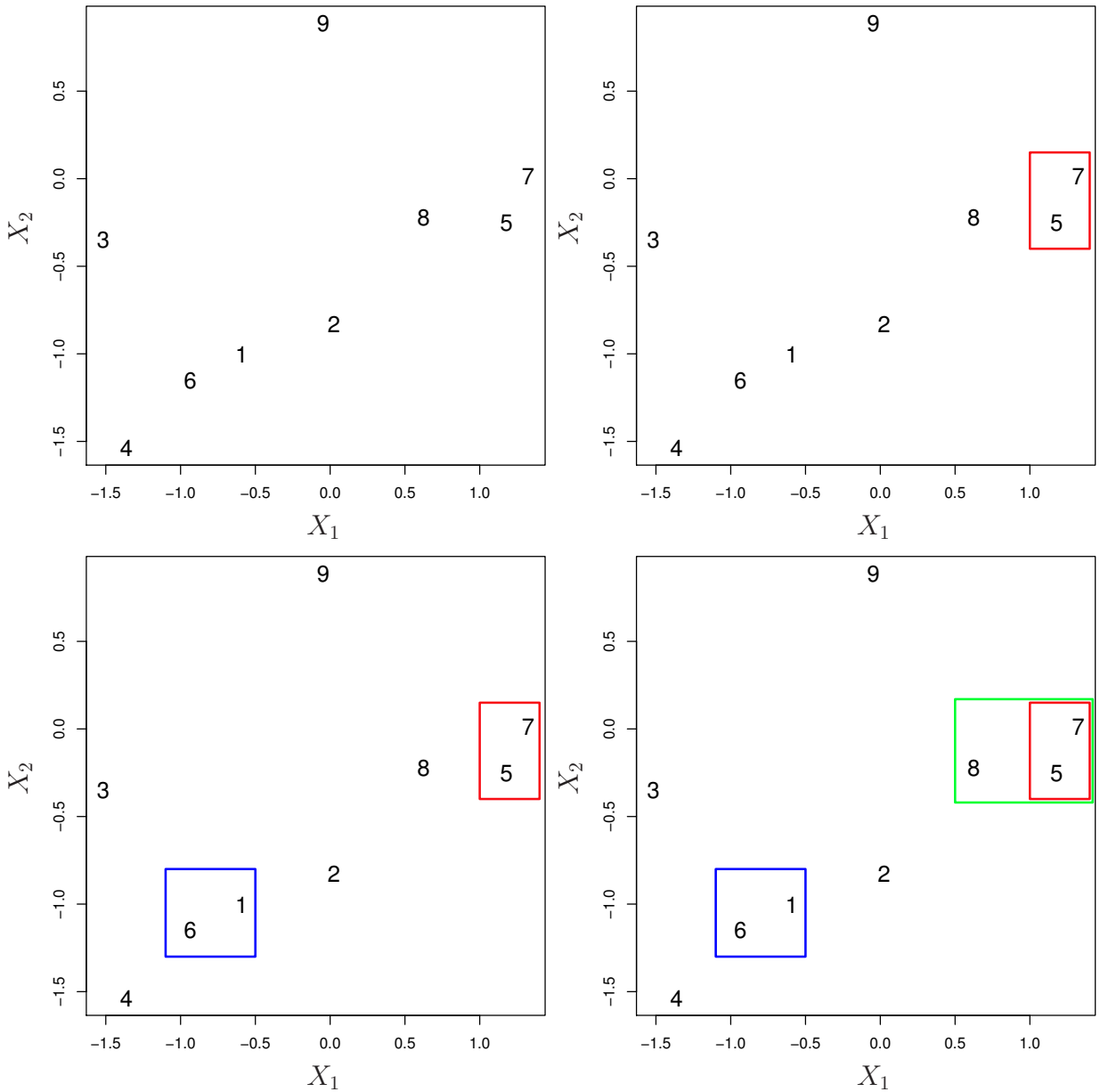

Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Example: Different Starting Values

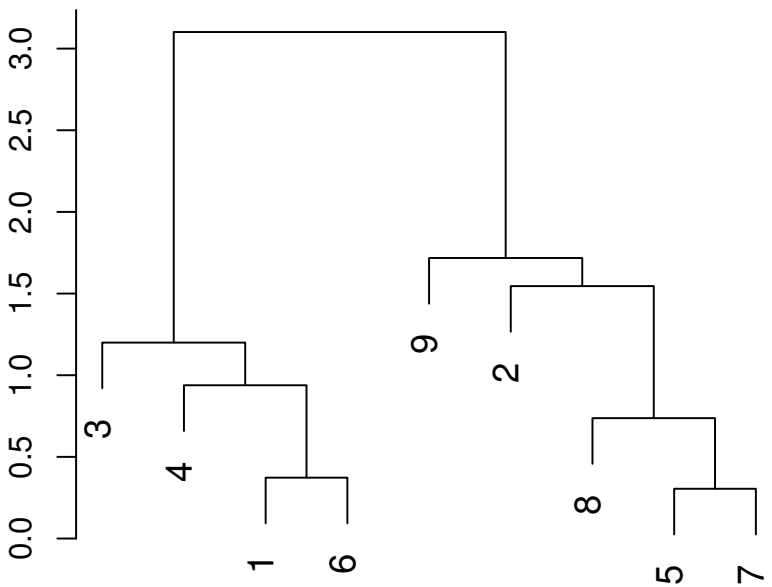
Hierarchical Clustering

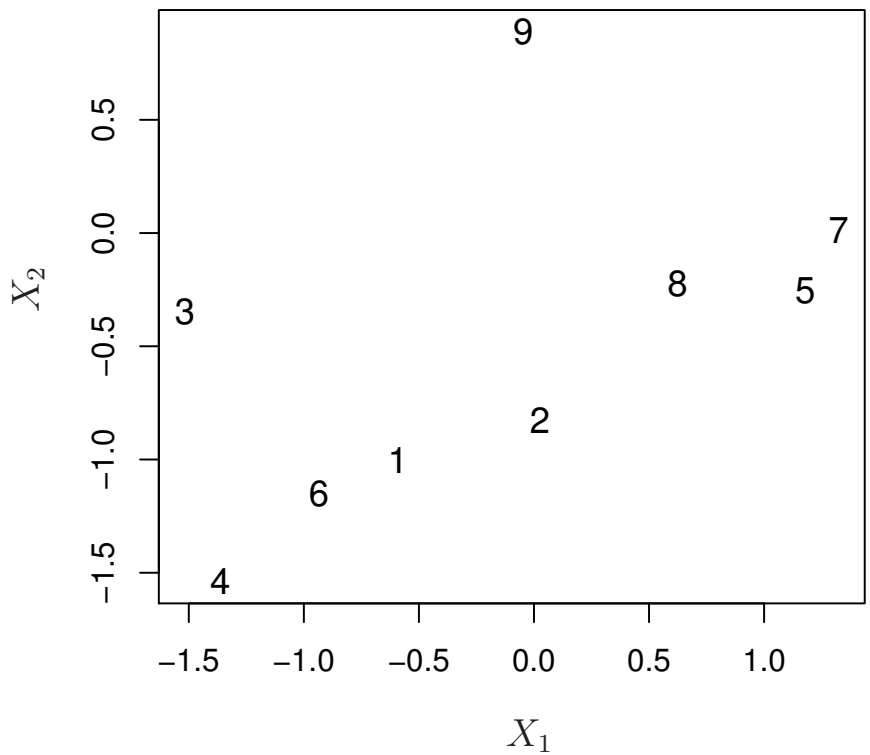
- K-means clustering requires us to pre-specify the number of clusters K. This can be a disadvantage (later we discuss strategies for choosing K)
- Hierarchical clustering is an alternative approach which does not require that we commit to a particular choice of K.
- In this section, we describe bottom-up or agglomerative clustering. This is the most common type of hierarchical clustering, and refers to the fact that a dendrogram is built starting from the leaves and combining clusters up to the trunk.

Hierarchical Clustering: The Idea



Hierarchical Clustering: The Idea

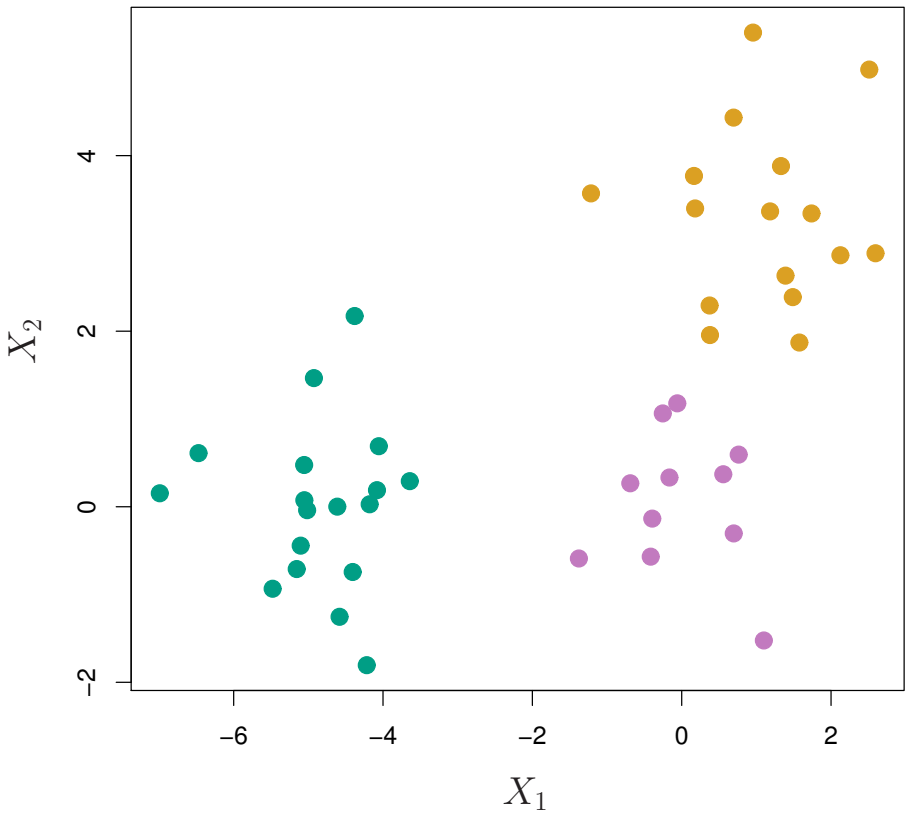




Types of Linkage

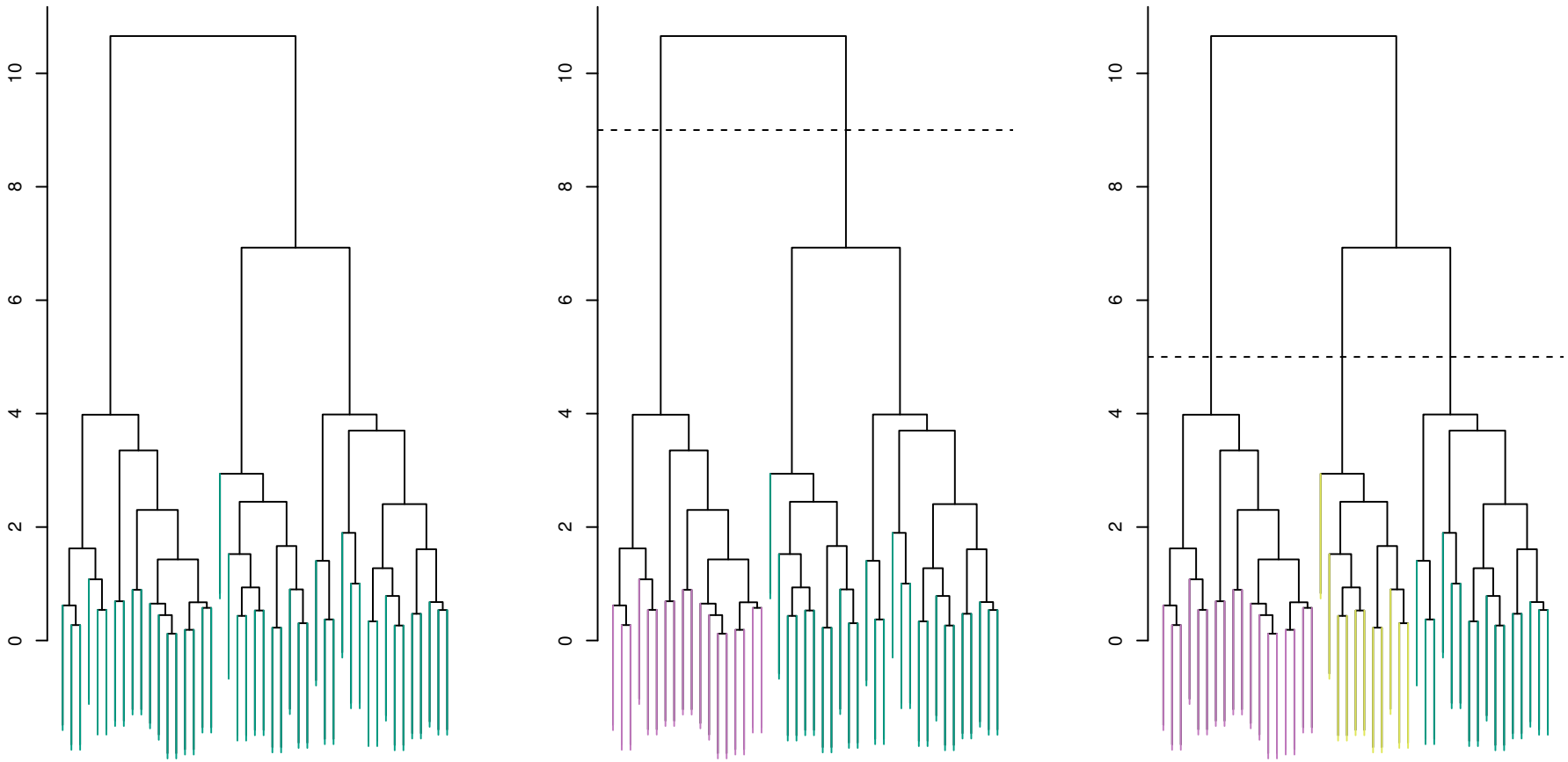
- Complete: Maximal inter-cluster dissimilarity.
  - Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the largest of these dissimilarities.
- Single: Minimal inter-cluster dissimilarity.
  - Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the smallest of these dissimilarities.
- Average: Mean inter-cluster dissimilarity.
  - Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the average of these dissimilarities.
- Centroid: Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable inversions. This occurs because the similarity measure is non-monotonic.

Example



- 45 observations generated in 2-dimensional space.
- In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown

Example



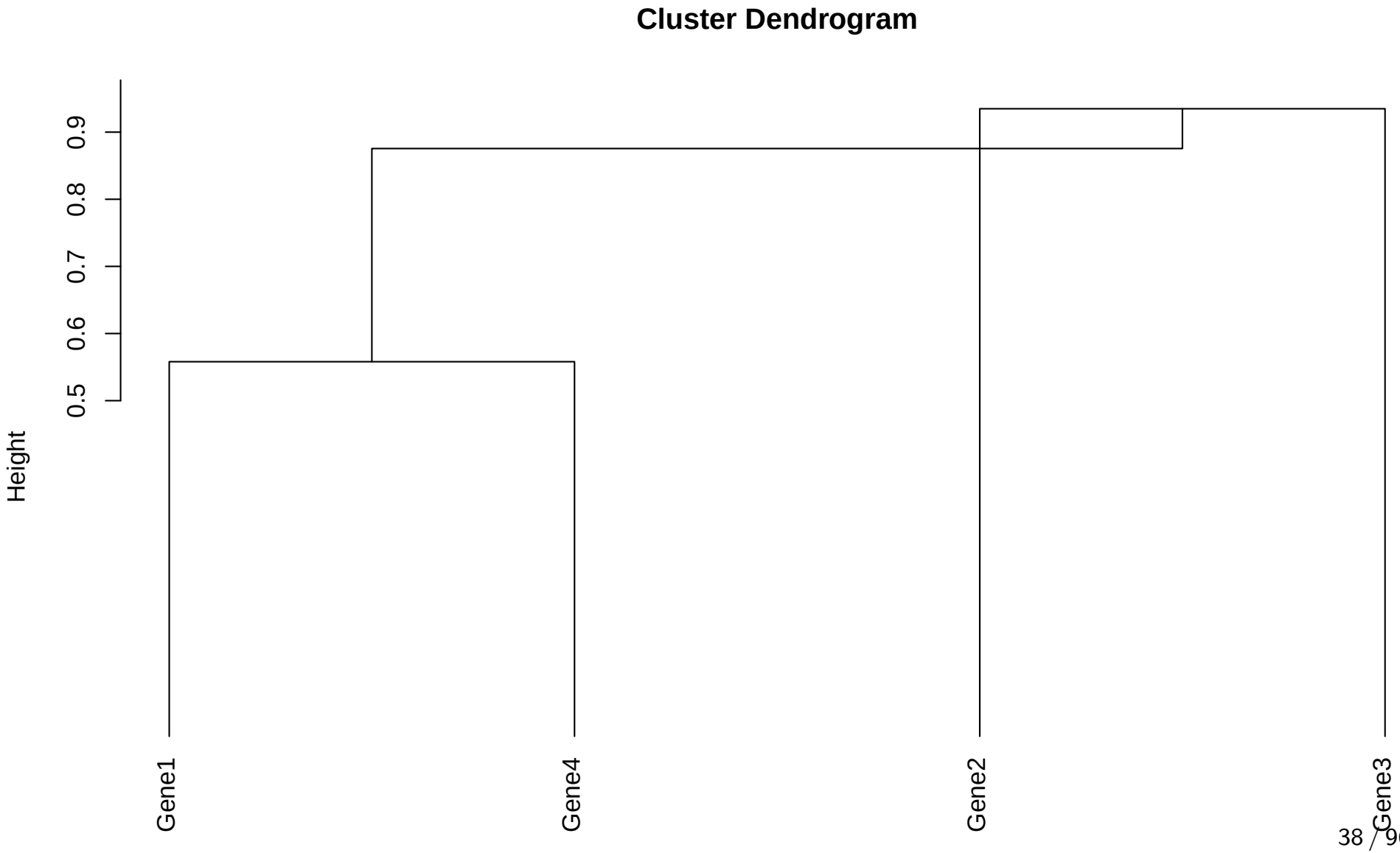
Example

- From http://www.sfu.ca/~bjonoska/STAT445/week9/inversionExample.r

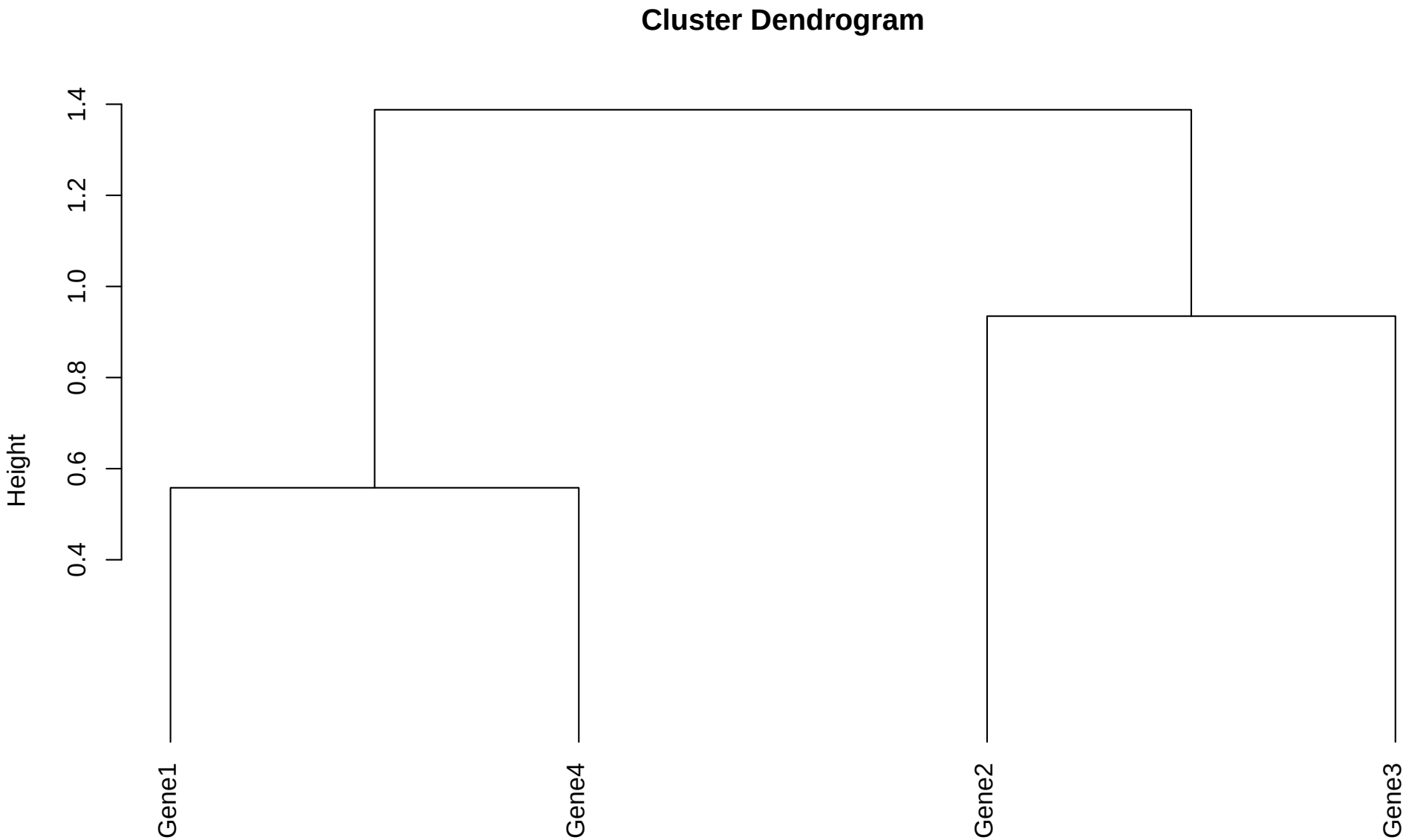
```
data  <- matrix(c(0.96, 0.07, 0.97, 0.98, 0.99, 0.50,
                  0.28, 0.29, 0.77, 0.78, 0.08, 0.96,
                  0.51, 0.51, 0.55, 0.14, 0.19, 0.41,
                  0.51, 0.40, 0.97, 0.98, 0.99, 0.50), ncol=6, byrow=TRUE)



colnames(data) <- c("Exp1","Exp2","Exp3","Exp4","Exp5","Exp6")
rownames(data) <- c("Gene1","Gene2","Gene3", "Gene4")

data
```

```
##       Exp1 Exp2 Exp3 Exp4 Exp5 Exp6
## Gene1 0.96 0.07 0.97 0.98 0.99 0.50
## Gene2 0.28 0.29 0.77 0.78 0.08 0.96
## Gene3 0.51 0.51 0.55 0.14 0.19 0.41
## Gene4 0.51 0.40 0.97 0.98 0.99 0.50
```

```r
mat <- dist(data, method="euclidean")
hc  <- hclust(mat, method='centroid')
plot(hc,hang=-1)
```

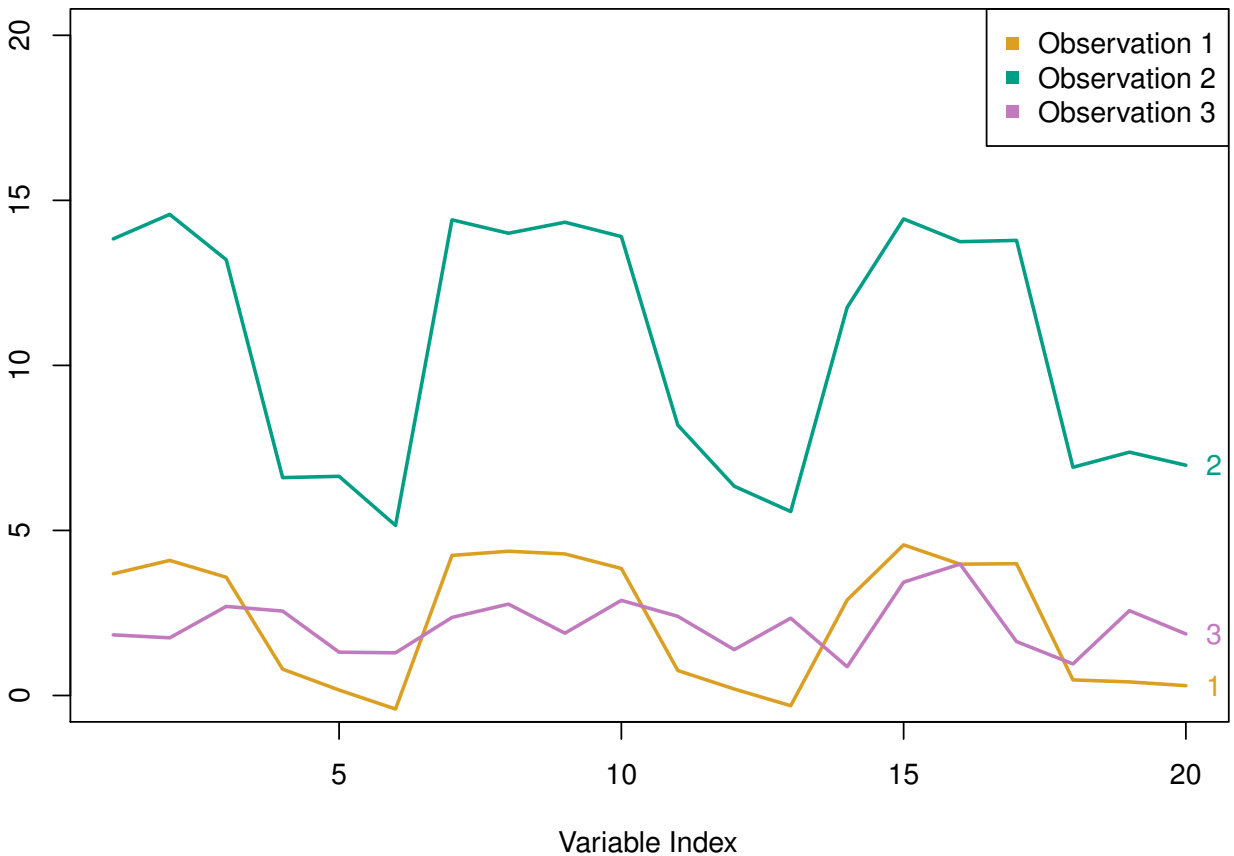


```r
hc <- hclust(mat, method='complete')
plot(hc,hang=-1)
```



Choice of Dissimilarity Measure

- So far have used Euclidean distance.
- An alternative is correlation-based distance which considers two observations to be similar if their features are highly correlated.
- This is an unusual use of correlation, which is normally computed between variables; here it is computed between the observation profiles for each pair of observations.



Practical Issues

- Should the observations or features first be standardized in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of hierarchical clustering,
  - What dissimilarity measure should be used?
  - What type of linkage should be used?
- How many clusters to choose? (in both K-means or hierarchical clustering). Difficult problem. No agreed-upon method. See *Elements of Statistical Learning*, chapter 13 for more details.

NCI60 Data

- Cancer cell line microarray data
- Consists of 6,830 gene expression measurements on 64 cancer cell lines.
- Each cell line is labeled with a cancer type. We won't use this for clustering purposes.

```
library(ISLR)
nci.labs <- NCI60$labs
nci.data <- NCI60$data
dim(nci.data)
```

```
## [1]   64 6830
```

```
nci.labs[1:4]
```

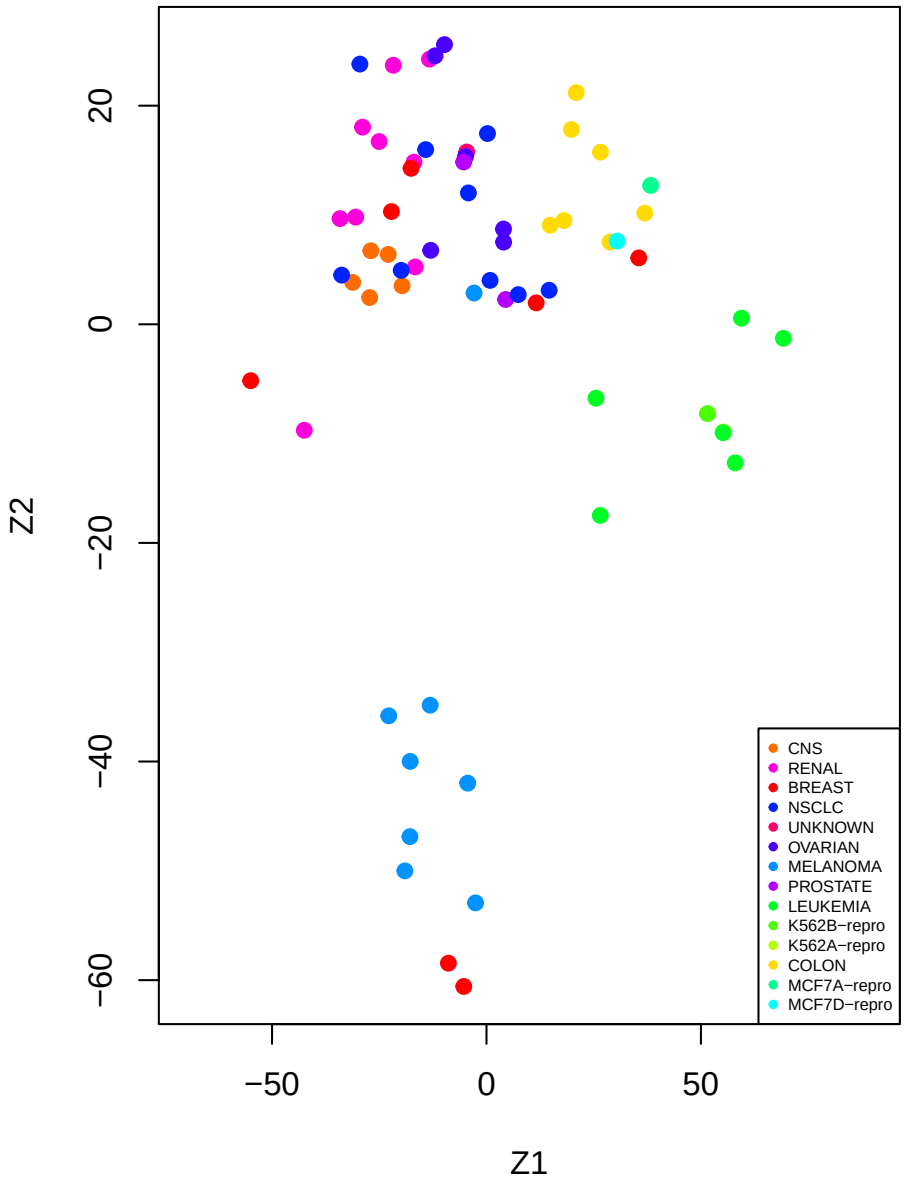
```
## [1] "CNS"   "CNS"   "CNS"   "RENAL"
```

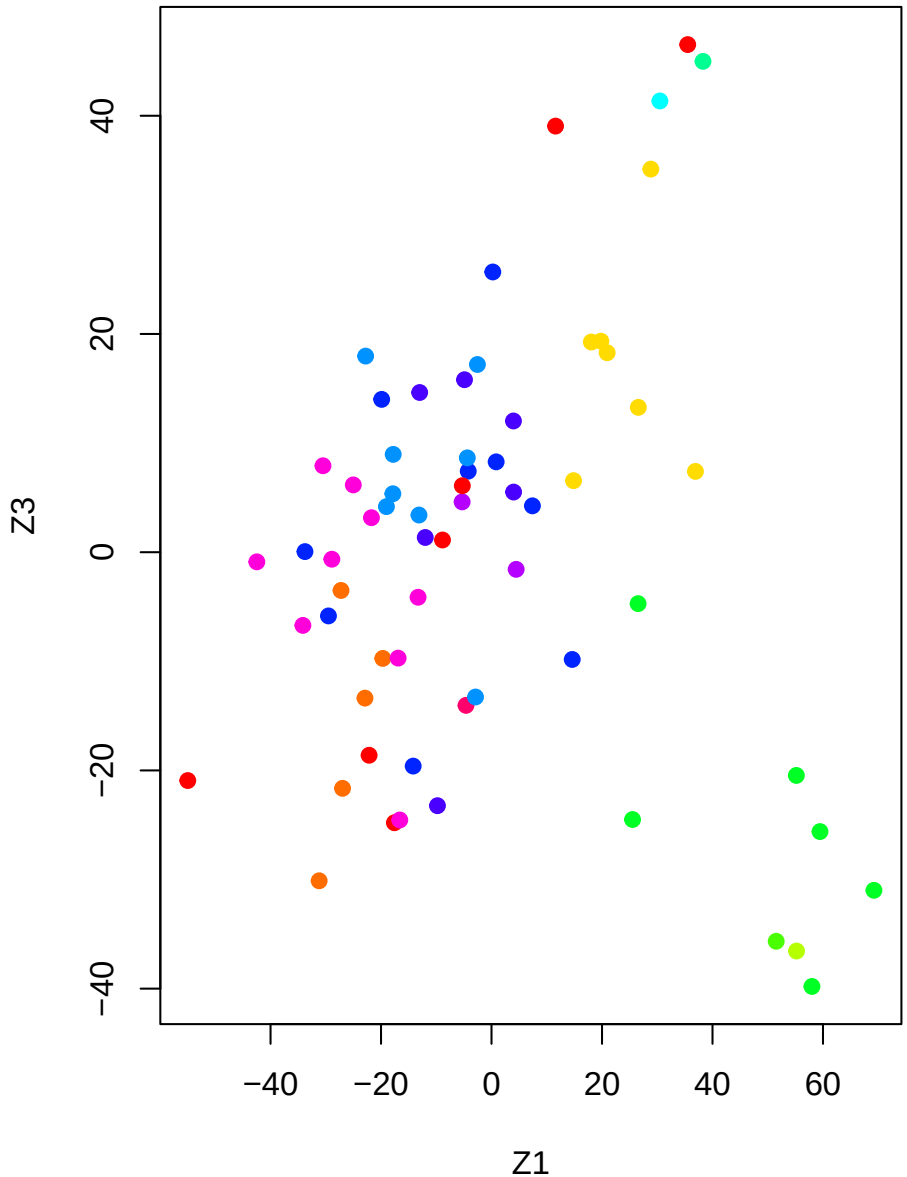
```
table(nci.labs)
```

```
## nci.labs
##      BREAST         CNS       COLON K562A-repro K562B-repro    LEUKEMIA
##           7           5           7           1           1           6
## MCF7A-repro MCF7D-repro    MELANOMA       NSCLC     OVARIAN    PROSTATE
##           1           1           8           9           6           2
##       RENAL     UNKNOWN
##           9           1
```

```
pr.out <- prcomp(nci.data, scale=TRUE)

Cols <- function(vec){
 cols <- rainbow (length (unique(vec)))
 return(cols[as.numeric(as.factor(vec))])
 }
```

```
par(mfrow=c(1,2))
plot(pr.out$x[,1:2], col=Cols(nci.labs), pch=19,
xlab ="Z1",ylab="Z2", xlim=c(-70, 90))
legend("bottomright", unique(nci.labs),
       pch=19, col=unique(Cols(nci.labs)), cex=0.5)

plot(pr.out$x[,c(1,3)], col=Cols(nci.labs), pch=19,
xlab ="Z1",ylab="Z3")
```


Z2
Z1
20
0
−20
−40
−60
−50
0
50
CNS
RENAL
BREAST
NSCLC
UNKNOWN
OVARIAN
MELANOMA
PROSTATE
LEUKEMIA
K562B−repro
K562A−repro
COLON
MCF7A−repro
MCF7D−repro


Z3
Z1
40
20
0
−20
−40
−40
−20
0
20
40
60

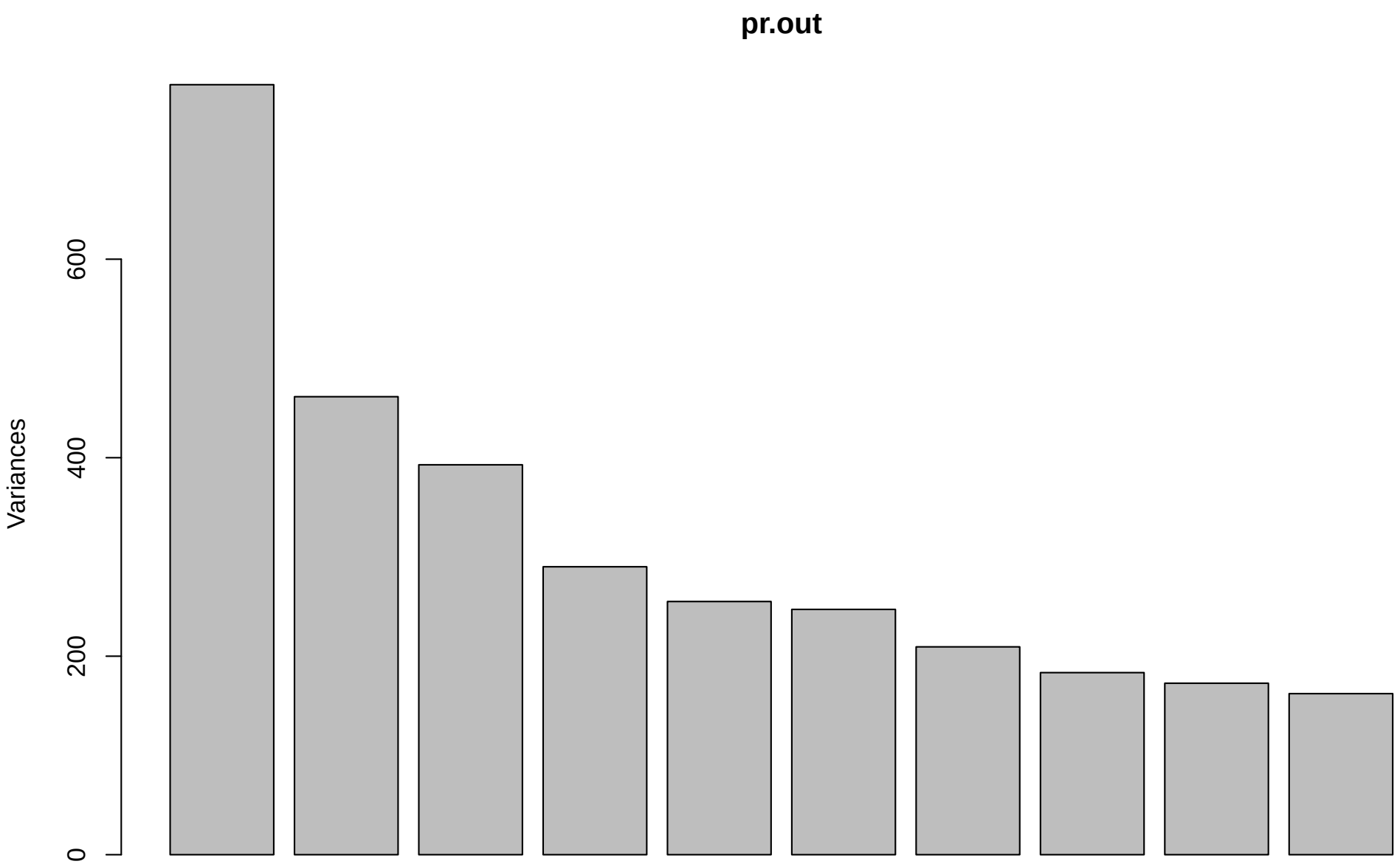
```
summary(pr.out)
```

```
## Importance of components:
##                             PC1      PC2      PC3      PC4      PC5      PC6
## Standard deviation     27.8535 21.48136 19.82046 17.03256 15.97181 15.72108
## Proportion of Variance  0.1136  0.06756  0.05752  0.04248  0.03735  0.03619
## Cumulative Proportion   0.1136  0.18115  0.23867  0.28115  0.31850  0.35468
##                             PC7      PC8      PC9     PC10     PC11     PC12
## Standard deviation     14.47145 13.54427 13.14400 12.73860 12.68672 12.15769
## Proportion of Variance  0.03066  0.02686  0.02529  0.02376  0.02357  0.02164
## Cumulative Proportion   0.38534  0.41220  0.43750  0.46126  0.48482  0.50646
##                            PC13     PC14     PC15     PC16     PC17     PC18
## Standard deviation     11.83019 11.62554 11.43779 11.00051 10.65666 10.48880
## Proportion of Variance  0.02049  0.01979  0.01915  0.01772  0.01663  0.01611
## Cumulative Proportion   0.52695  0.54674  0.56590  0.58361  0.60024  0.61635
##                            PC19    PC20     PC21    PC22    PC23    PC24
## Standard deviation     10.43518 10.3219 10.14608 10.0544 9.90265 9.64766
## Proportion of Variance  0.01594  0.0156  0.01507  0.0148 0.01436 0.01363
## Cumulative Proportion   0.63229  0.6479  0.66296  0.6778 0.69212 0.70575
##                           PC25    PC26    PC27   PC28    PC29    PC30    PC31
## Standard deviation     9.50764 9.33253 9.27320 9.0900 8.98117 8.75003 8.59962
## Proportion of Variance 0.01324 0.01275 0.01259 0.0121 0.01181 0.01121 0.01083
## Cumulative Proportion  0.71899 0.73174 0.74433 0.7564 0.76824 0.77945 0.79027
##                           PC32    PC33    PC34    PC35    PC36    PC37    PC38
## Standard deviation     8.44738 8.37305 8.21579 8.15731 7.97465 7.90446 7.82127
## Proportion of Variance 0.01045 0.01026 0.00988 0.00974 0.00931 0.00915 0.00896
## Cumulative Proportion  0.80072 0.81099 0.82087 0.83061 0.83992 0.84907 0.85803
##                           PC39    PC40    PC41   PC42    PC43   PC44    PC45
## Standard deviation     7.72156 7.58603 7.45619 7.3444 7.10449 7.0131 6.95839
## Proportion of Variance 0.00873 0.00843 0.00814 0.0079 0.00739 0.0072 0.00709
## Cumulative Proportion  0.86676 0.87518 0.88332 0.8912 0.89861 0.9058 0.91290
##                           PC46    PC47    PC48    PC49    PC50    PC51    PC52
## Standard deviation     6.8663 6.80744 6.64763 6.61607 6.40793 6.21984 6.20326
## Proportion of Variance 0.0069 0.00678 0.00647 0.00641 0.00601 0.00566 0.00563
## Cumulative Proportion  0.9198 0.92659 0.93306 0.93947 0.94548 0.95114 0.95678
##                           PC53    PC54    PC55    PC56    PC57   PC58    PC59
## Standard deviation     6.06706 5.91805 5.91233 5.73539 5.47261 5.2921 5.02117
```

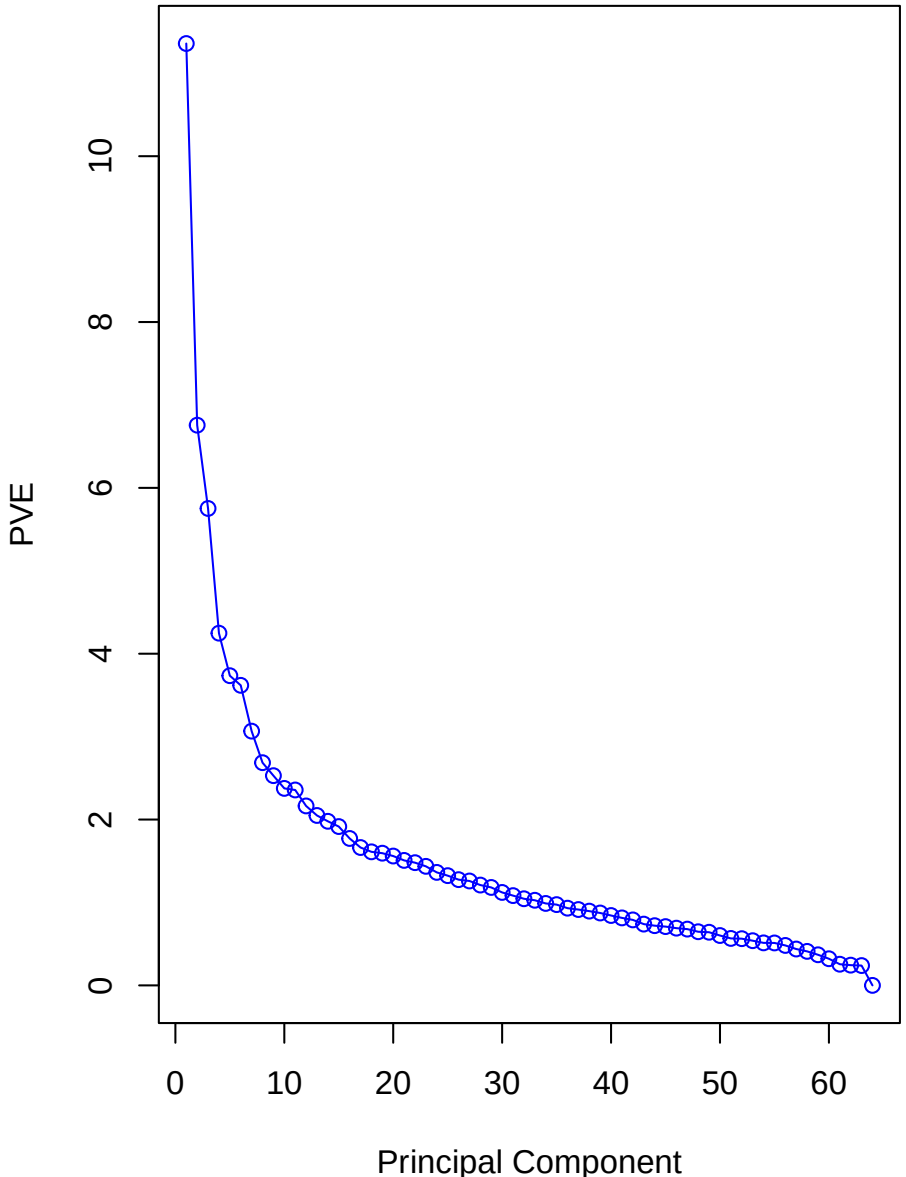
```
pr.out$sdev^2
```

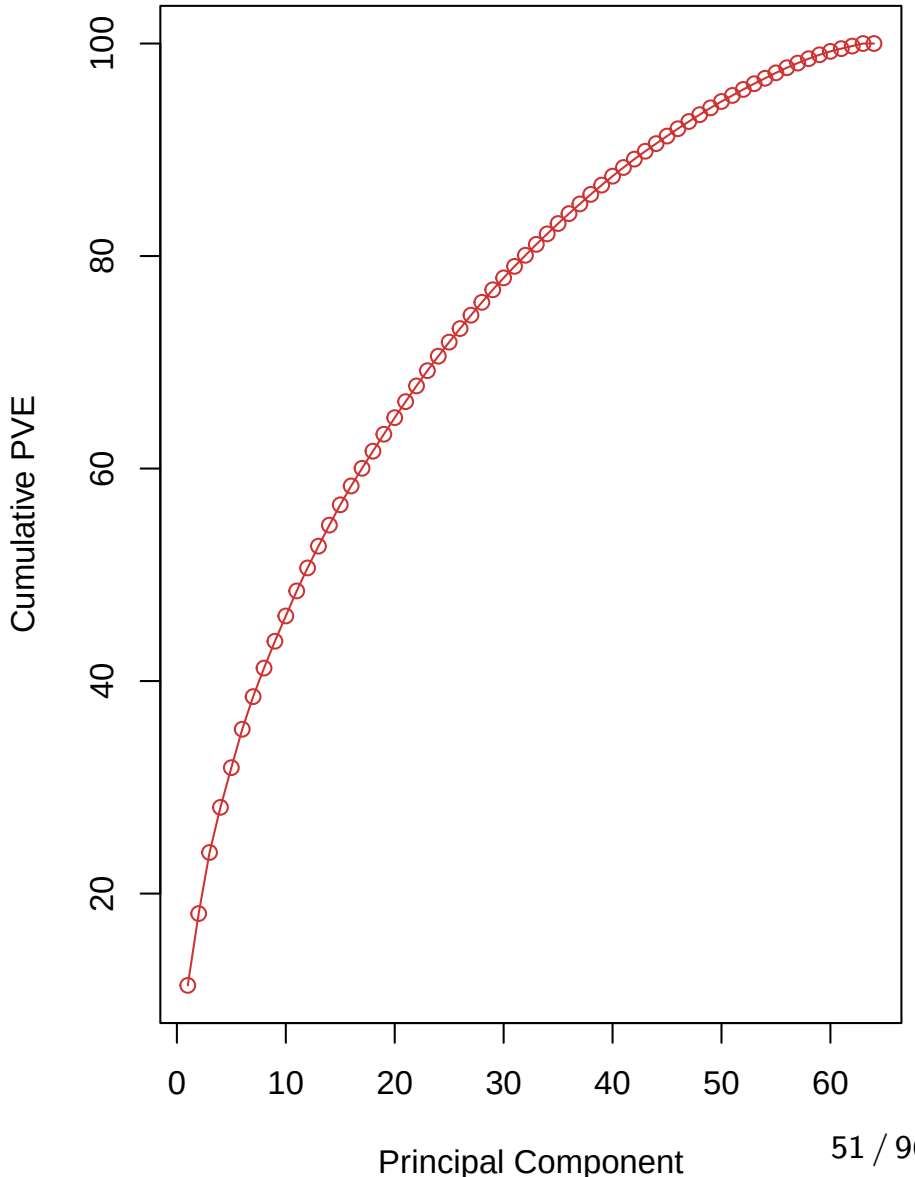
```
##  [1] 7.758157e+02 4.614486e+02 3.928508e+02 2.901080e+02 2.550986e+02
##  [6] 2.471524e+02 2.094230e+02 1.834472e+02 1.727647e+02 1.622718e+02
## [11] 1.609529e+02 1.478095e+02 1.399534e+02 1.351533e+02 1.308230e+02
## [16] 1.210113e+02 1.135644e+02 1.100148e+02 1.088930e+02 1.065424e+02
## [21] 1.029429e+02 1.010908e+02 9.806257e+01 9.307726e+01 9.039519e+01
## [26] 8.709610e+01 8.599223e+01 8.262894e+01 8.066147e+01 7.656305e+01
## [31] 7.395349e+01 7.135815e+01 7.010794e+01 6.749915e+01 6.654176e+01
## [36] 6.359512e+01 6.248052e+01 6.117227e+01 5.962252e+01 5.754792e+01
## [41] 5.559482e+01 5.393991e+01 5.047377e+01 4.918294e+01 4.841912e+01
## [46] 4.714560e+01 4.634123e+01 4.419098e+01 4.377236e+01 4.106152e+01
## [51] 3.868639e+01 3.848041e+01 3.680928e+01 3.502331e+01 3.495568e+01
## [56] 3.289465e+01 2.994946e+01 2.800683e+01 2.521219e+01 2.193966e+01
## [61] 1.743625e+01 1.666371e+01 1.633165e+01 4.614964e-28
```

```
plot(pr.out)
```



```r
pve <- 100*pr.out$sdev^2/sum(pr.out$sdev ^2)
par(mfrow=c(1,2))
plot(pve, type ="o", ylab="PVE", xlab="Principal Component",
col ="blue")
plot(cumsum(pve), type="o", ylab = "Cumulative PVE",
xlab="Principal Component", col= "brown3")
```

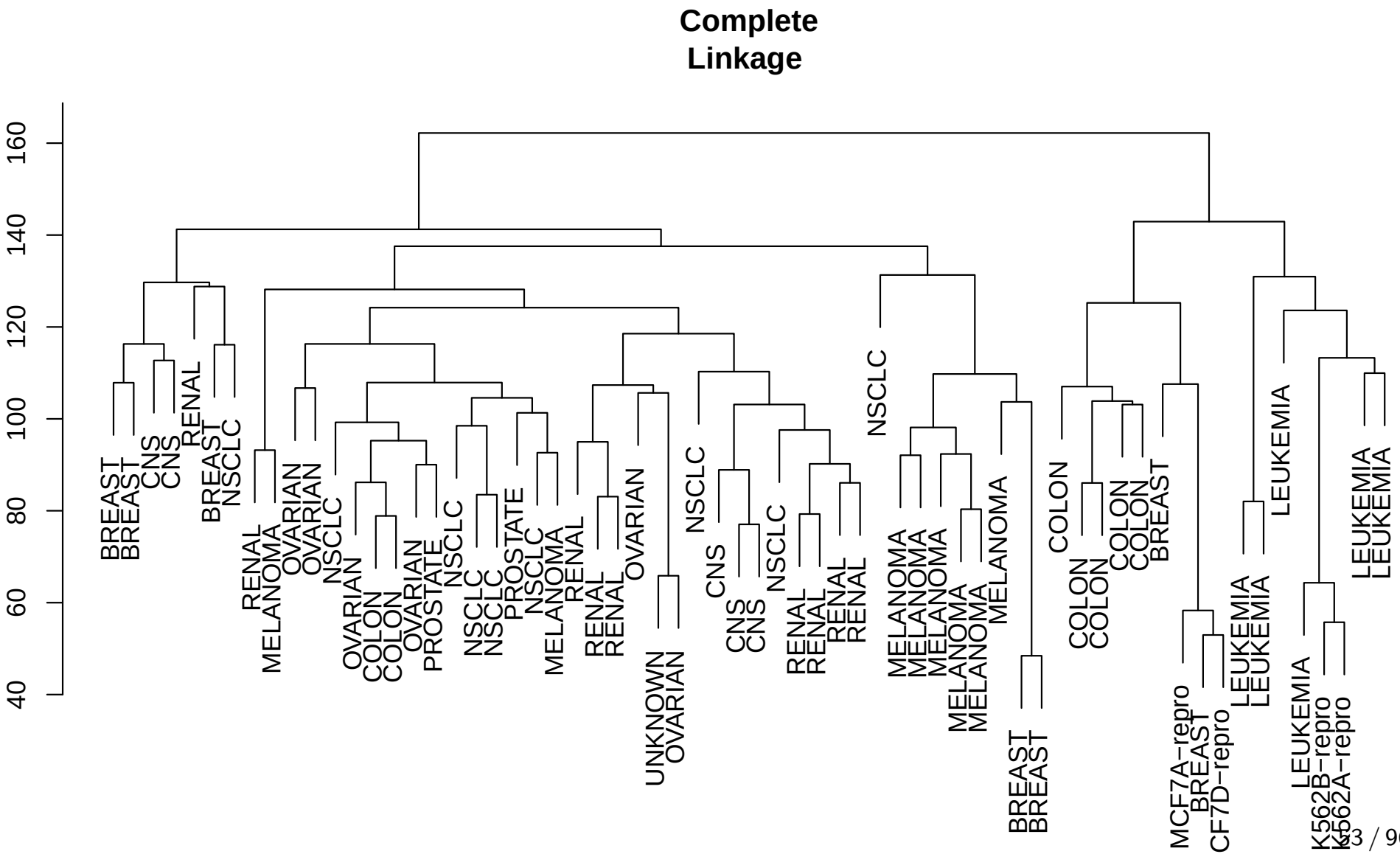




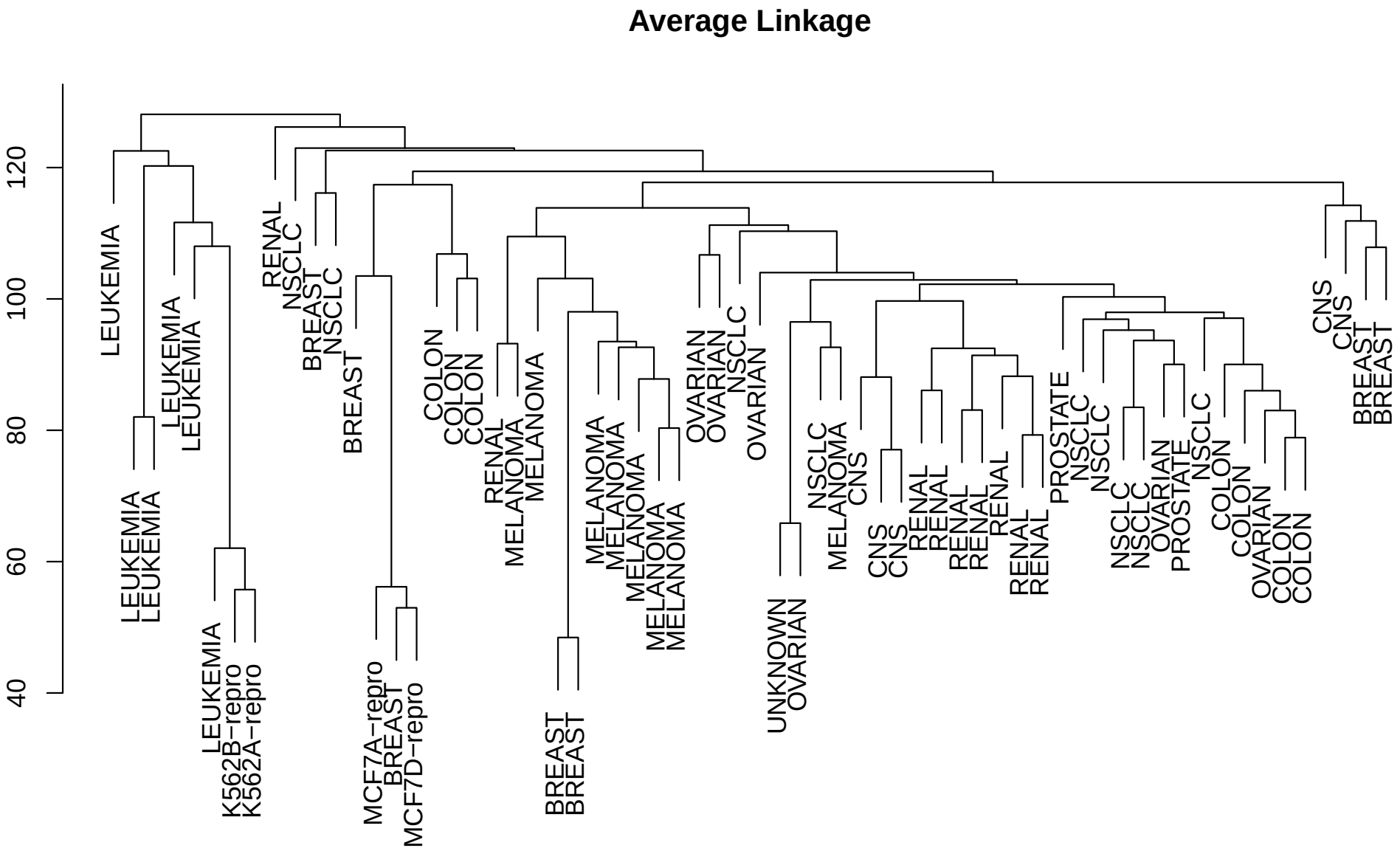
- Standardize the variables to have mean zero and standard deviation one.

```
sd.data <- scale(nci.data)
```

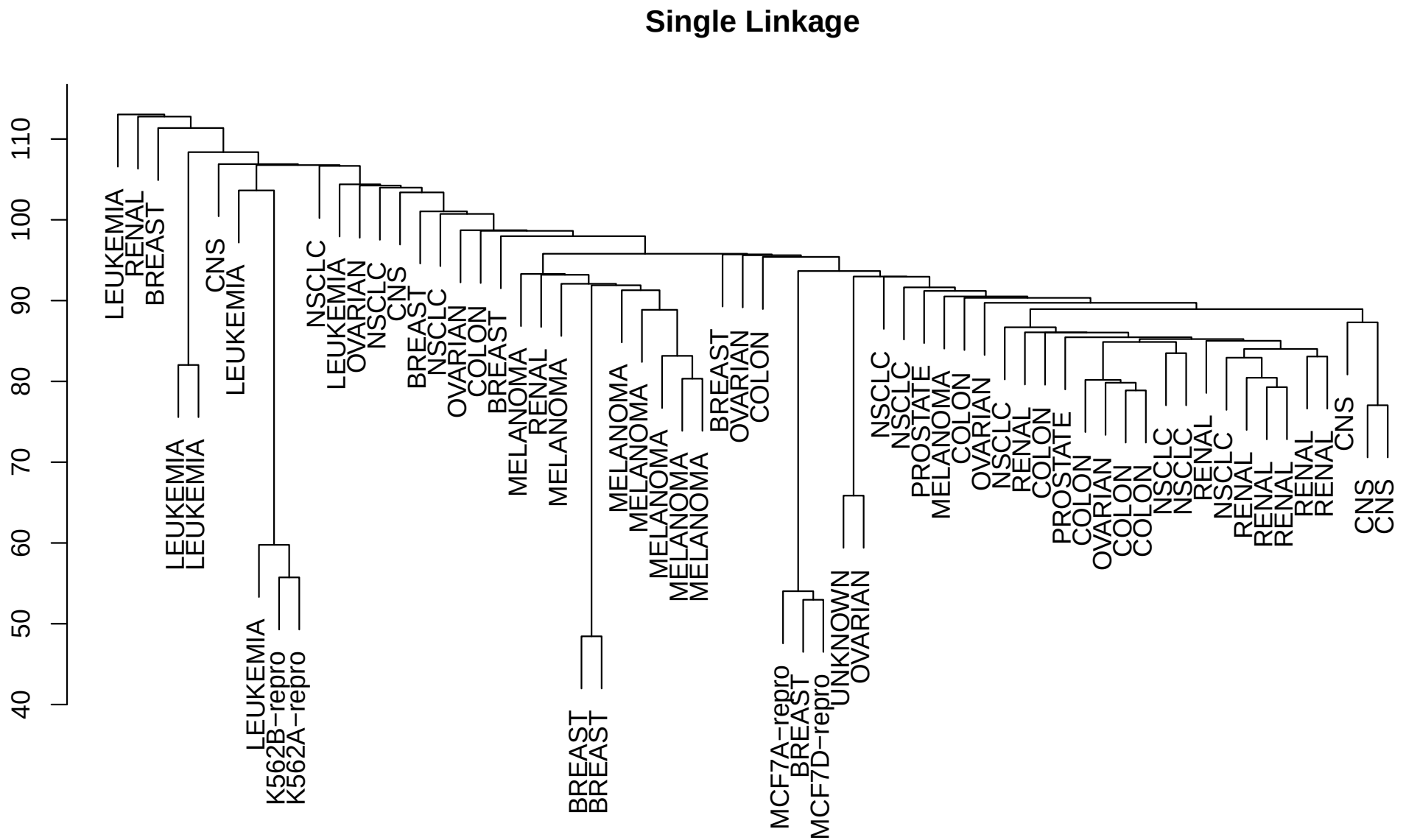
```
data.dist <- dist(sd.data)
plot(hclust(data.dist), labels=nci.labs, main="Complete
Linkage ", xlab ="", sub ="", ylab ="")
```



```
plot(hclust(data.dist, method ="average"), labels=nci.labs,
main="Average Linkage", xlab ="", sub ="", ylab ="")
```



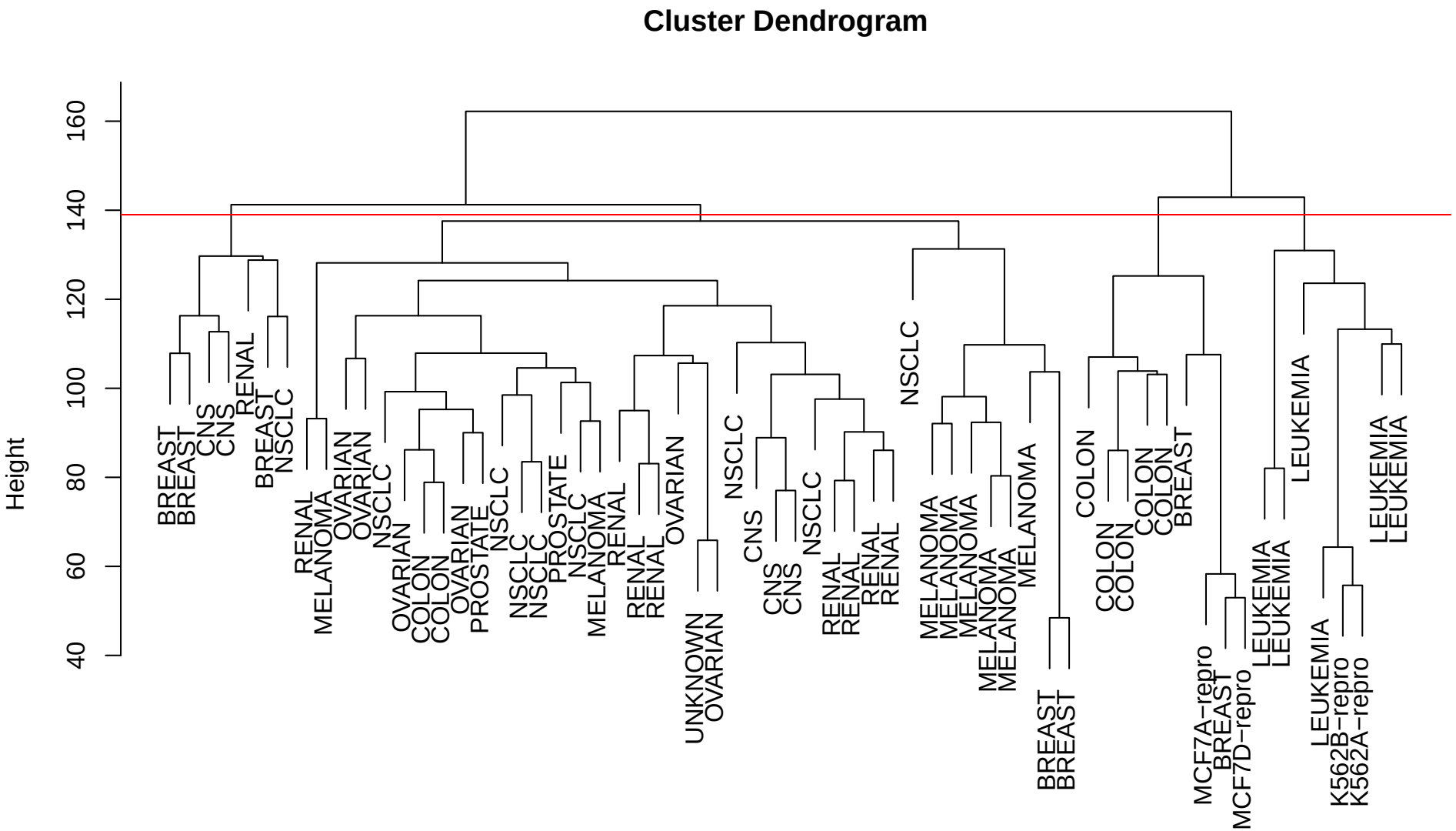
```
plot(hclust(data.dist, method ="single"), labels=nci.labs,
main="Single Linkage", xlab="", sub="", ylab="")
```



```
hc.out <- hclust(dist(sd.data))
hc.clusters <- cutree(hc.out ,4)
table(hc.clusters, nci.labs)
```

```
##             nci.labs
## hc.clusters BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro
##           1      2   3     2           0           0        0           0
##           2      3   2     0           0           0        0           0
##           3      0   0     0           1           1        6           0
##           4      2   0     5           0           0        0           1
##             nci.labs
## hc.clusters MCF7D-repro MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
##           1           0        8     8       6        2     8       1
##           2           0        0     1       0        0     1       0
##           3           0        0     0       0        0     0       0
##           4           1        0     0       0        0     0       0
```

```
plot(hc.out, labels=nci.labs)
abline(h=139, col="red")
```



```
hc.out
```

```
##
## Call:
## hclust(d = dist(sd.data))
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 64
```

```
set.seed(2)
km.out <- kmeans(sd.data, 4, nstart=20)
km.clusters <- km.out$cluster
names(km.clusters) <- nci.labs
```

```
names(km.clusters[km.clusters==1])
```

```
##  [1] "PROSTATE"    "OVARIAN"     "OVARIAN"     "OVARIAN"     "NSCLC"
##  [6] "NSCLC"       "COLON"       "COLON"       "COLON"       "COLON"
## [11] "COLON"       "COLON"       "COLON"       "MCF7A-repro" "BREAST"
## [16] "MCF7D-repro" "BREAST"      "NSCLC"       "NSCLC"       "NSCLC"
```

```
names(km.clusters[km.clusters==2])
```

```
##  [1] "CNS"      "CNS"      "CNS"      "RENAL"    "BREAST"   "CNS"
##  [7] "CNS"      "BREAST"   "NSCLC"    "NSCLC"    "RENAL"    "RENAL"
## [13] "RENAL"    "RENAL"    "RENAL"    "RENAL"    "RENAL"    "BREAST"
## [19] "NSCLC"    "RENAL"    "UNKNOWN"  "OVARIAN"  "MELANOMA" "OVARIAN"
## [25] "OVARIAN"  "PROSTATE" "NSCLC"
```

```
names(km.clusters[km.clusters==3])
```

```
## [1] "MELANOMA" "BREAST"   "BREAST"   "MELANOMA" "MELANOMA" "MELANOMA" "MELANOMA"
## [8] "MELANOMA" "MELANOMA"
```

```
names(km.clusters[km.clusters==4])
```

```
## [1] "LEUKEMIA"    "K562B-repro" "K562A-repro" "LEUKEMIA"    "LEUKEMIA"
## [6] "LEUKEMIA"    "LEUKEMIA"    "LEUKEMIA"
```

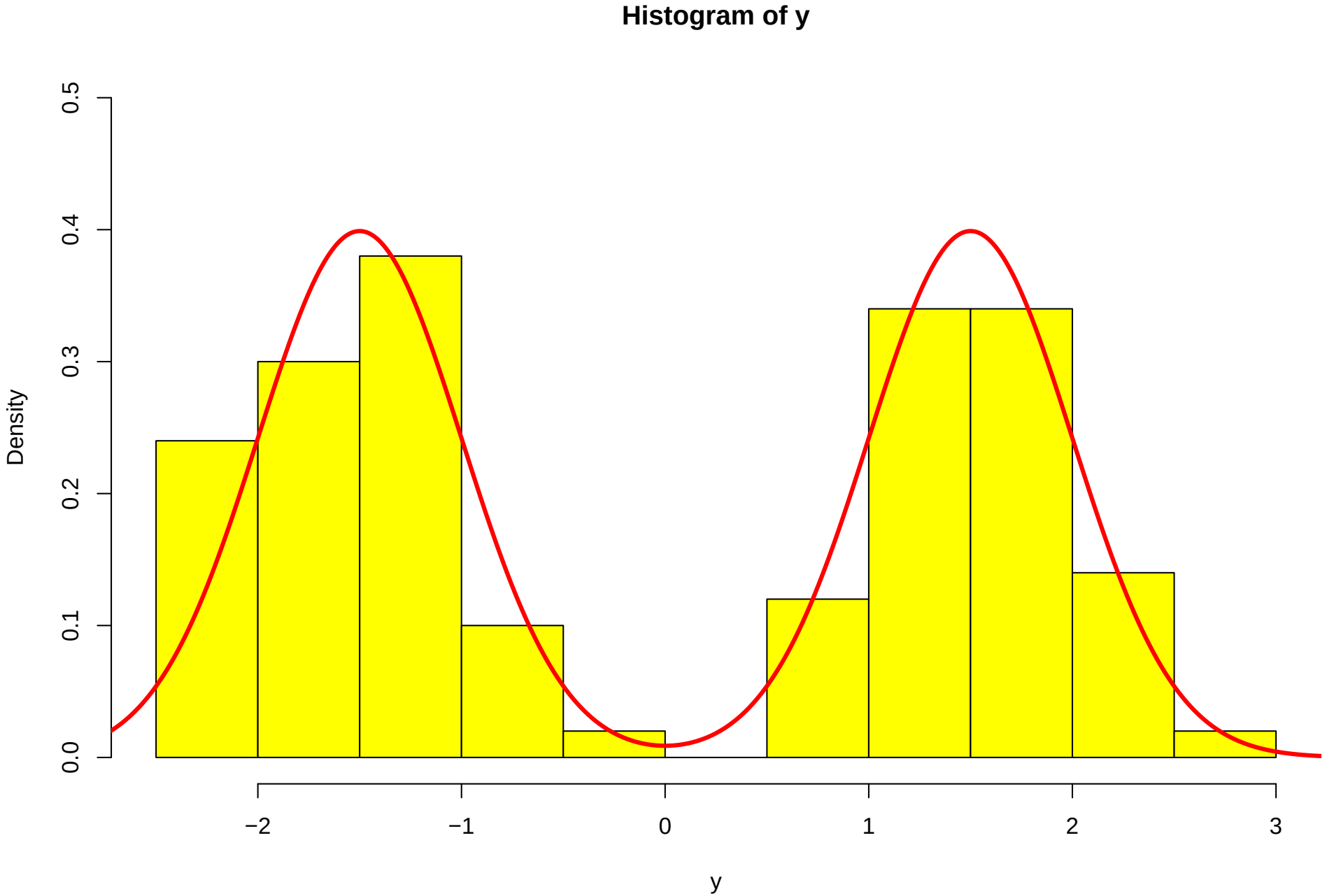
Model Based Clustering

- Let's start simply. Let's generate data from the following model:

$$y_i = \left(1 - \frac{1}{2}\right) \; N\left[-\frac{3}{2}, \left(\frac{1}{2}\right)^2\right] \; + \; \frac{1}{2} \; N\left[\frac{3}{2}, \left(\frac{1}{2}\right)^2\right]$$

```r
set.seed(1001)
n <- 100
z <- rbinom(n, 1, 1/2)

y <- rep(0, n)
y[z==0] <- rnorm(length(y[z==0]), -3/2, 1/2)
y[z==1] <- rnorm(length(y[z==1]), 3/2, 1/2)
```

```r
hist(y, col="yellow", prob=TRUE, ylim=c(0, 0.5))
x <- seq(-4,4, by=0.01)
lines(x, 0.5*dnorm(x, -3/2, 1/2) +
        0.5*dnorm(x, 3/2, 1/2), lwd=3, col="red")
```


Histogram of y
Density
0.0
0.1
0.2
0.3
0.4
0.5
−2
−1
0
1
2
3
y

mclust

```
library("mclust")
```

```
## Package 'mclust' version 5.4.8
## Type 'citation("mclust")' for citing this R package in publications.
```

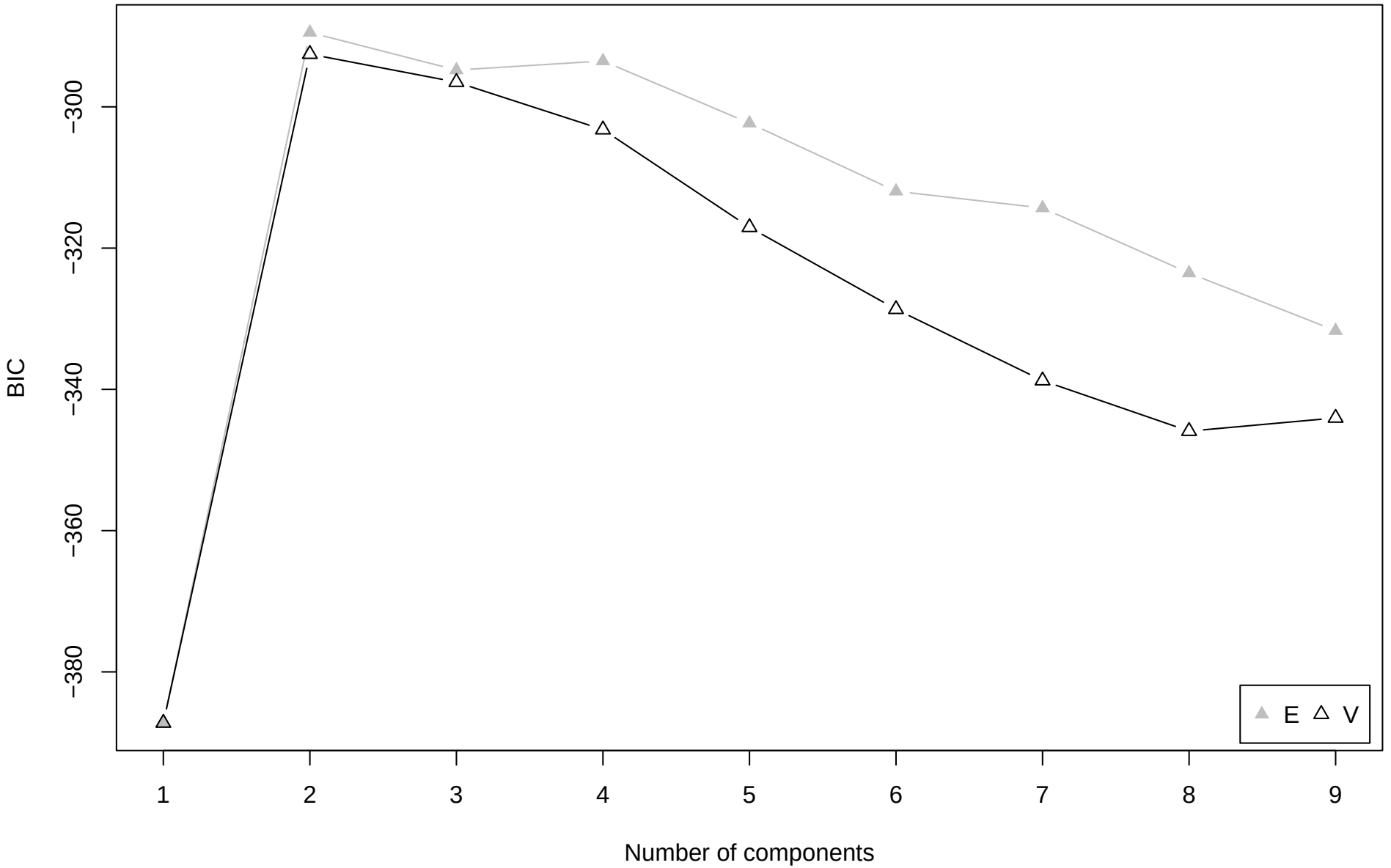
```
citation("mclust")
```

```
##
## To cite 'mclust' R package in publications, please use:
##
##   Scrucca L., Fop M., Murphy T. B. and Raftery A. E. (2016) mclust 5:
##   clustering, classification and density estimation using Gaussian
##   finite mixture models The R Journal 8/1, pp. 289-317
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {{mclust} 5: clustering, classification and density estimati
##     author = {Luca Scrucca and Michael Fop and T. Brendan Murphy and Adr
##     journal = {The {R} Journal},
##     year = {2016},
##     volume = {8},
```

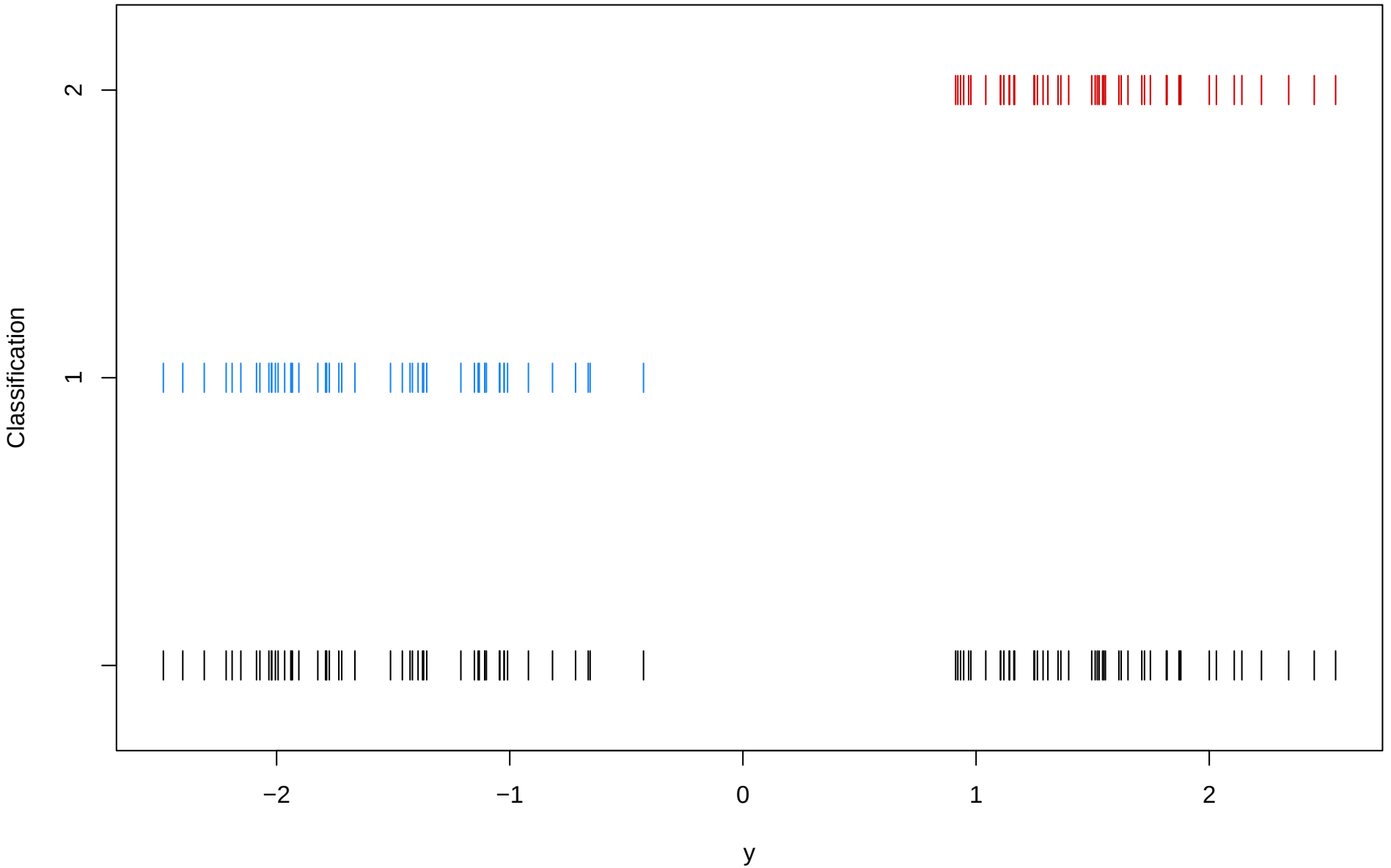
```r
mod <- Mclust(y)
summary(mod, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust E (univariate, equal variance) model with 2 components:
##
##  log-likelihood   n df       BIC       ICL
##       -135.5219 100  4 -289.4644 -289.4709
##
## Clustering table:
##  1  2
## 52 48
##
## Mixing probabilities:
##         1         2
## 0.5199668 0.4800332
##
## Means:
##         1         2
## -1.545128  1.520145
##
## Variances:
##         1         2
## 0.2205437 0.2205437
```

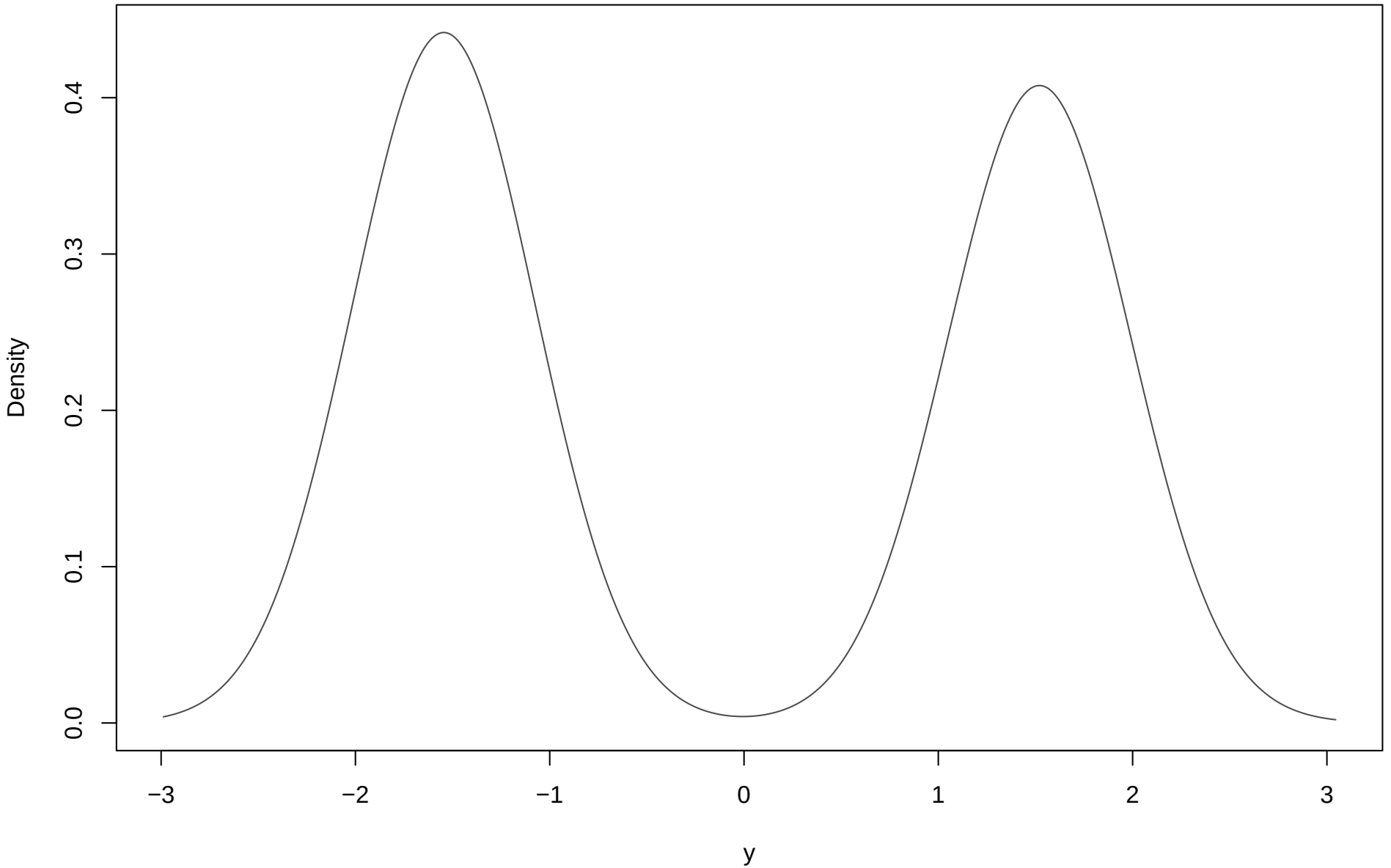
```
plot(mod, what = "BIC", main = FALSE)
```



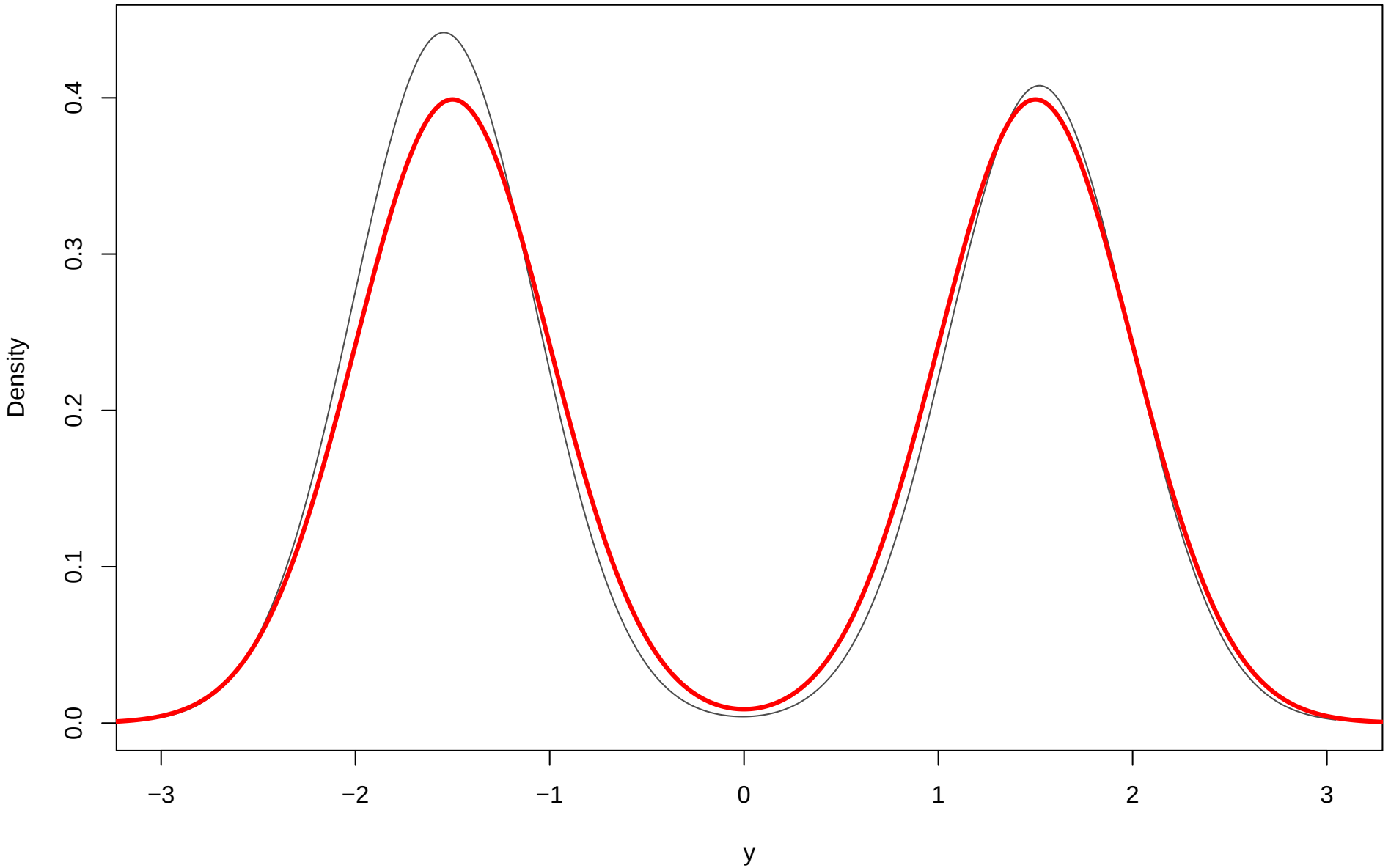
```
plot(mod, what = "classification", main = FALSE)
```



```
plot(mod, what = "density", main = FALSE)
```



```r
plot(mod, what = "density", main = FALSE)
x <- seq(-4,4, by=0.01)
lines(x, 0.5*dnorm(x, -3/2, 1/2) +
        0.5*dnorm(x, 3/2, 1/2), lwd=3, col="red")
```


Density
0.0
0.1
0.2
0.3
0.4
−3
−2
−1
0
1
2
3
y

Bootstrap

```
boot <- MclustBootstrap(mod, nboot = 999, type = "bs")
summary(boot, what = "ci")
```

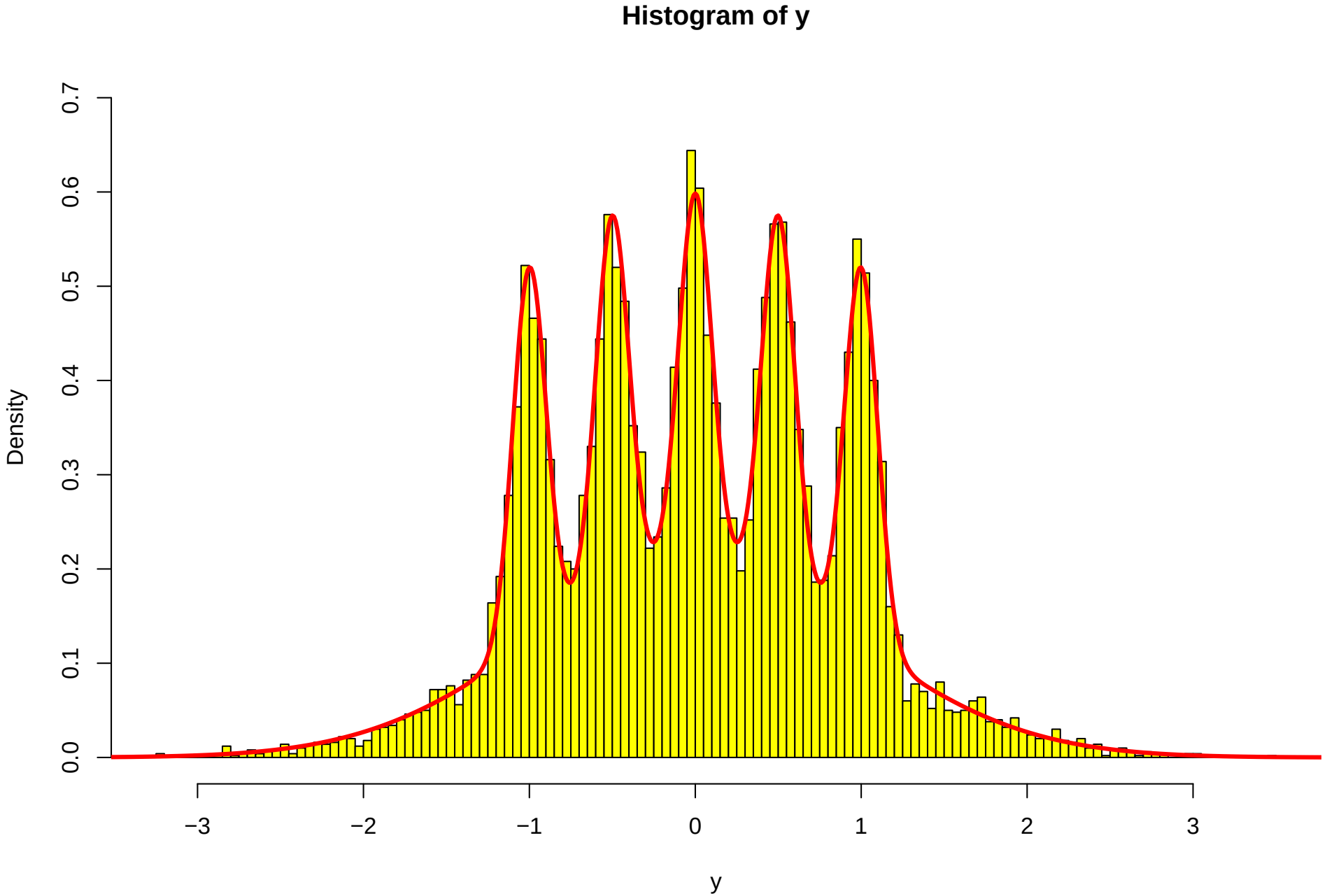
```
## ---------------------------------------------------------
## Resampling confidence intervals
## ---------------------------------------------------------
## Model                        = E
## Num. of mixture components = 2
## Replications                 = 999
## Type                         = nonparametric bootstrap
## Confidence level             = 0.95
##
## Mixing probabilities:
##                1         2
## 2.5%  0.4199981 0.3800277
## 97.5% 0.6199723 0.5800019
##
## Means:
##                1        2
## 2.5%  -1.683074 1.399398
## 97.5% -1.402782 1.640314
##
## Variances:
##                1         2
## 2.5%  0.1691538 0.1691538
## 97.5% 0.2639207 0.2639207
```

A Little More Complicated - The Claw

$$y_i = \frac{1}{2}\ N\left[0,1\right]\ +\ \sum_{i=0}^{4} \frac{1}{10}\ N\left[\frac{i}{2}-1, \left(\frac{1}{10}\right)^2\right]$$

```r
set.seed(2001)
n <- 10000
z <- sample(6, n, prob=c(1/2, 1/10, 1/10, 1/10, 1/10, 1/10),
            replace=TRUE)


y <- rep(0, n)
y[z==1] <- rnorm(length(y[z==1]), 0, 1)
y[z==2] <- rnorm(length(y[z==2]), 0/2-1, 1/10)
y[z==3] <- rnorm(length(y[z==3]), 1/2-1, 1/10)
y[z==4] <- rnorm(length(y[z==4]), 2/2-1, 1/10)
y[z==5] <- rnorm(length(y[z==5]), 3/2-1, 1/10)
y[z==6] <- rnorm(length(y[z==6]), 4/2-1, 1/10)
```

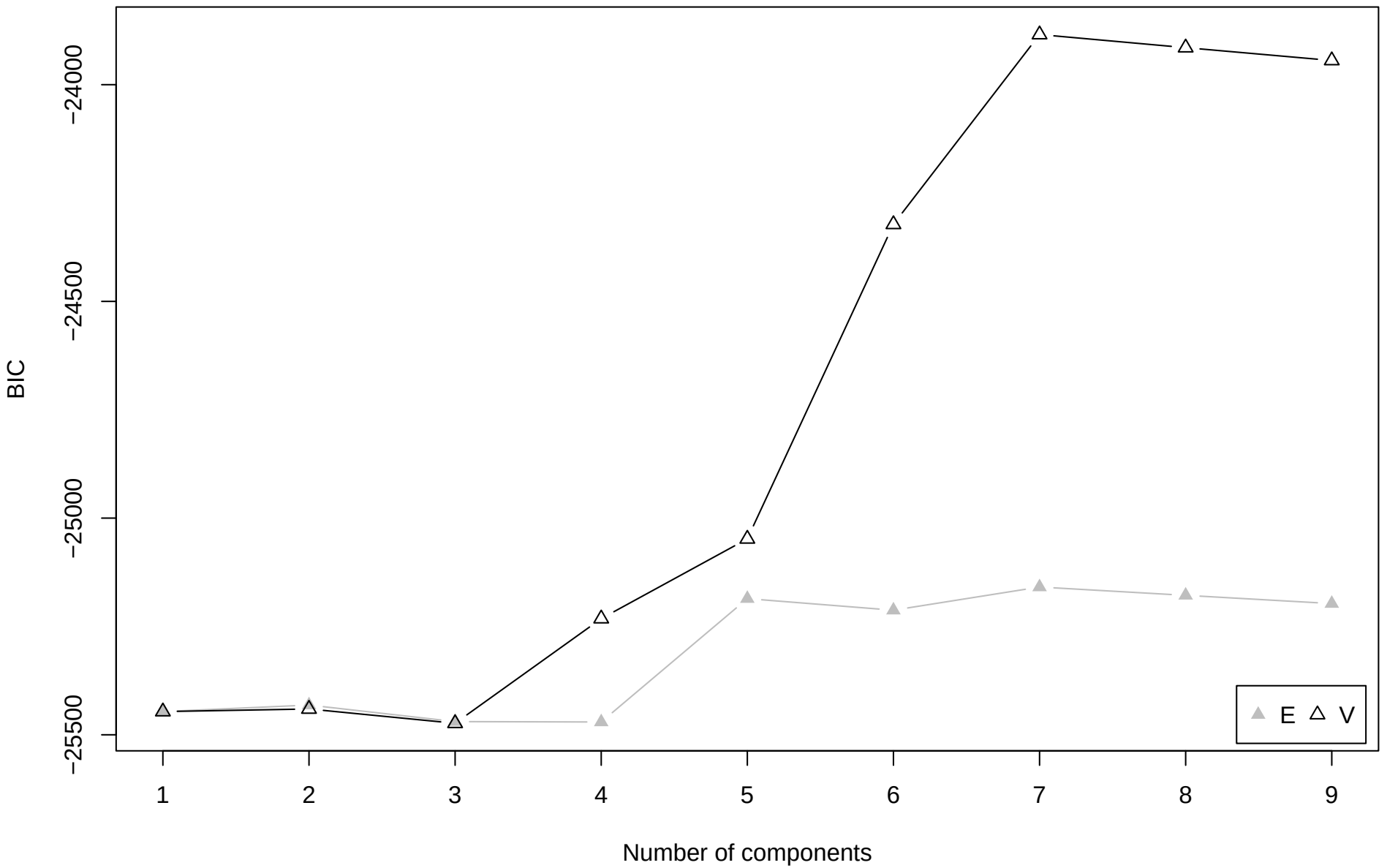
```
hist(y, col="yellow", prob=TRUE, ylim=c(0, 0.7), nclass=100)
x <- seq(-4,4, by=0.01)
lines(x, 0.5*dnorm(x, 0, 1) +
        (1/10)*dnorm(x, 0/2-1, 1/10) +
        (1/10)*dnorm(x, 1/2-1, 1/10) +
        (1/10)*dnorm(x, 2/2-1, 1/10) +
        (1/10)*dnorm(x, 3/2-1, 1/10) +
        (1/10)*dnorm(x, 4/2-1, 1/10), lwd=3, col="red")
```


Histogram of y
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
−3
−2
−1
0
1
2
3
y

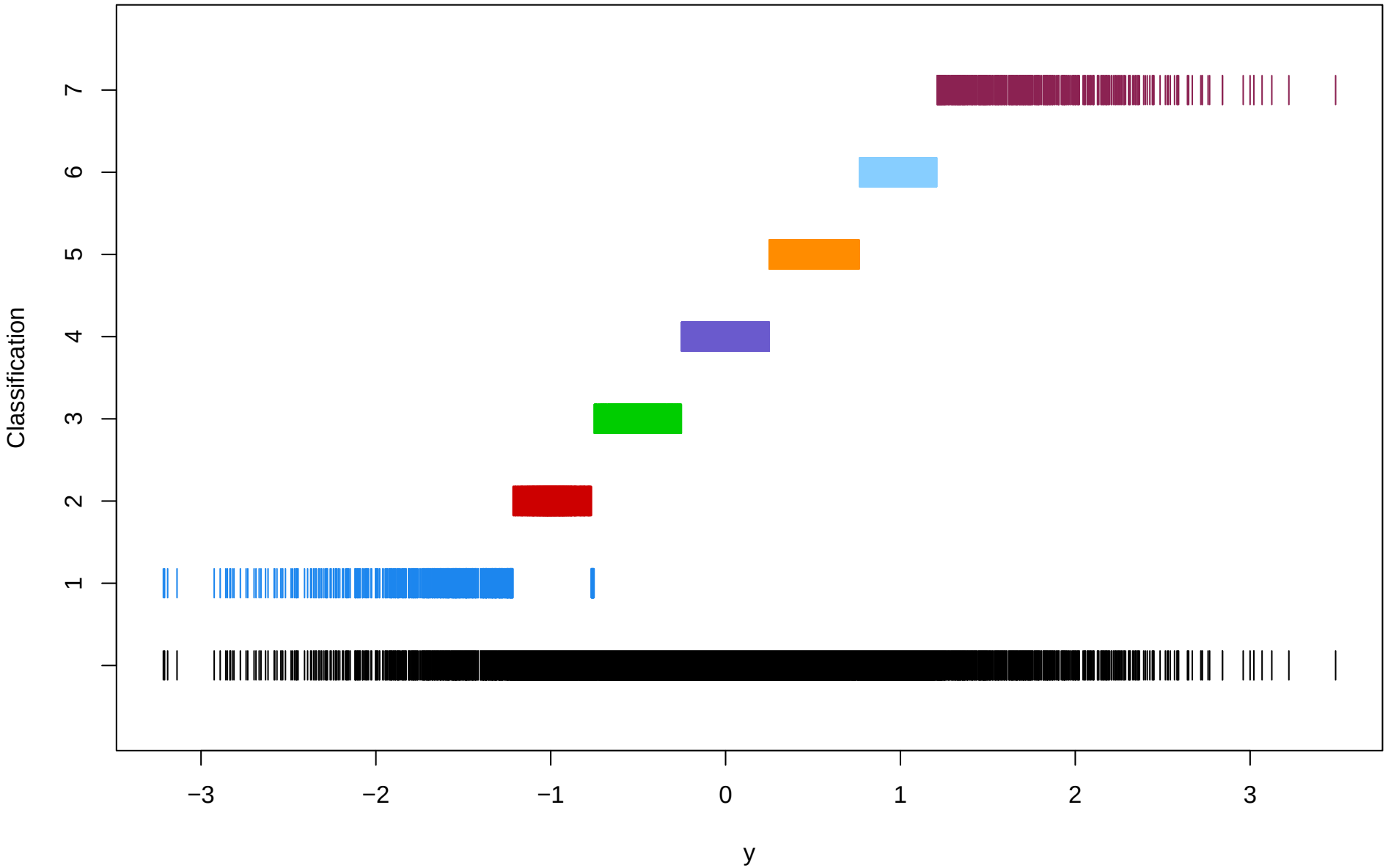
```r
mod <- Mclust(y)
summary(mod, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust V (univariate, unequal variance) model with 7 components:
##
##  log-likelihood     n df       BIC       ICL
##       -11850.07 10000 20 -23884.34 -28002.01
##
## Clustering table:
##    1    2    3    4    5    6    7
##  603 1503 1861 2012 1906 1547  568
##
## Mixing probabilities:
##         1         2         3         4         5         6         7
## 0.1296462 0.1226258 0.1591970 0.1805717 0.1684883 0.1294437 0.1100272
##
## Means:
##            1            2            3            4            5            6
## -1.044829890 -0.993289236 -0.491941895 -0.004012648  0.496700845  0.989874790
##            7
##  1.179194737
##
## Variances:
##          1          2          3          4          5          6          7
## 0.47294845 0.01414987 0.01715648 0.01671273 0.01642080 0.01209269 0.44469182
```

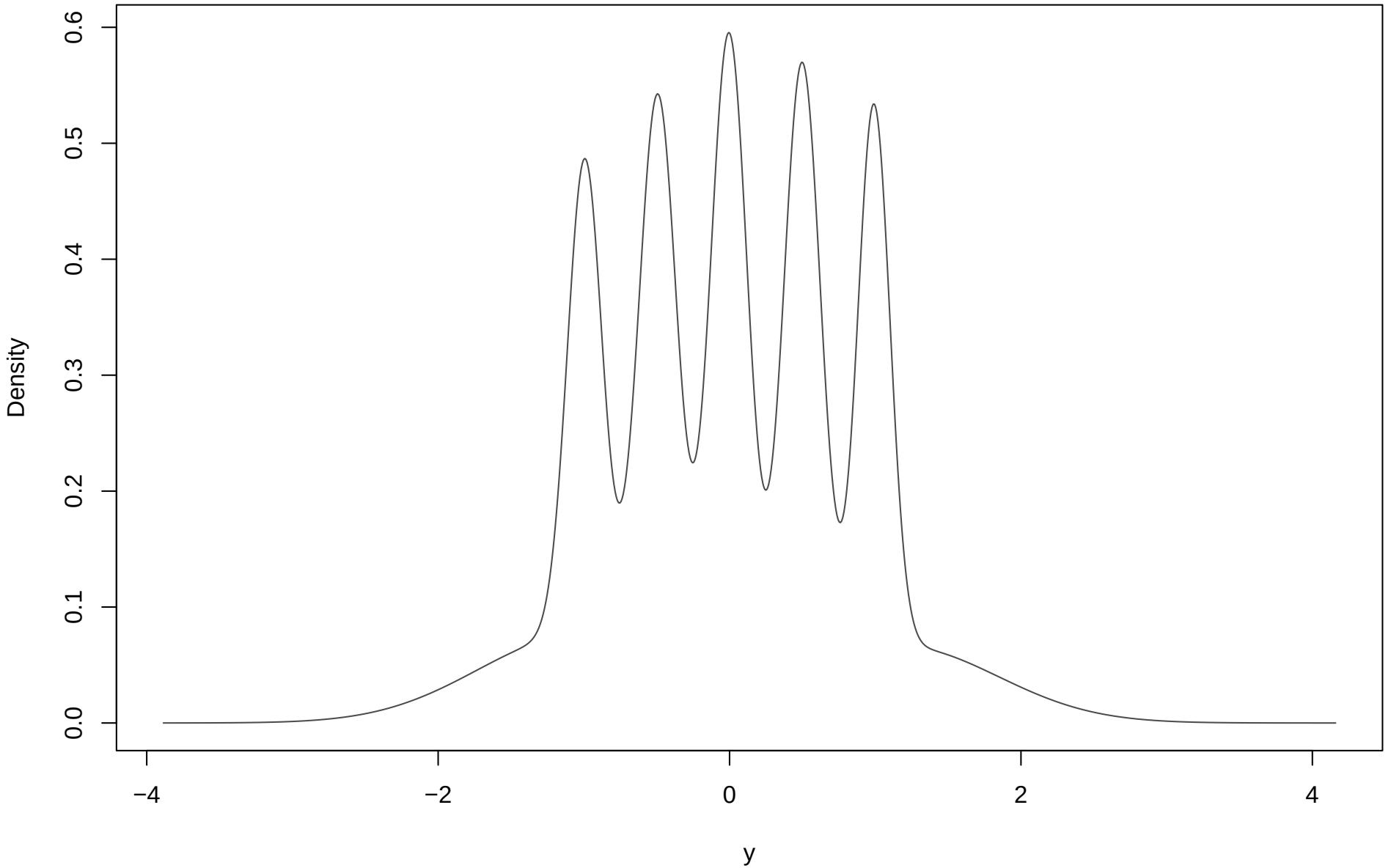
```
plot(mod, what = "BIC", main = FALSE)
```

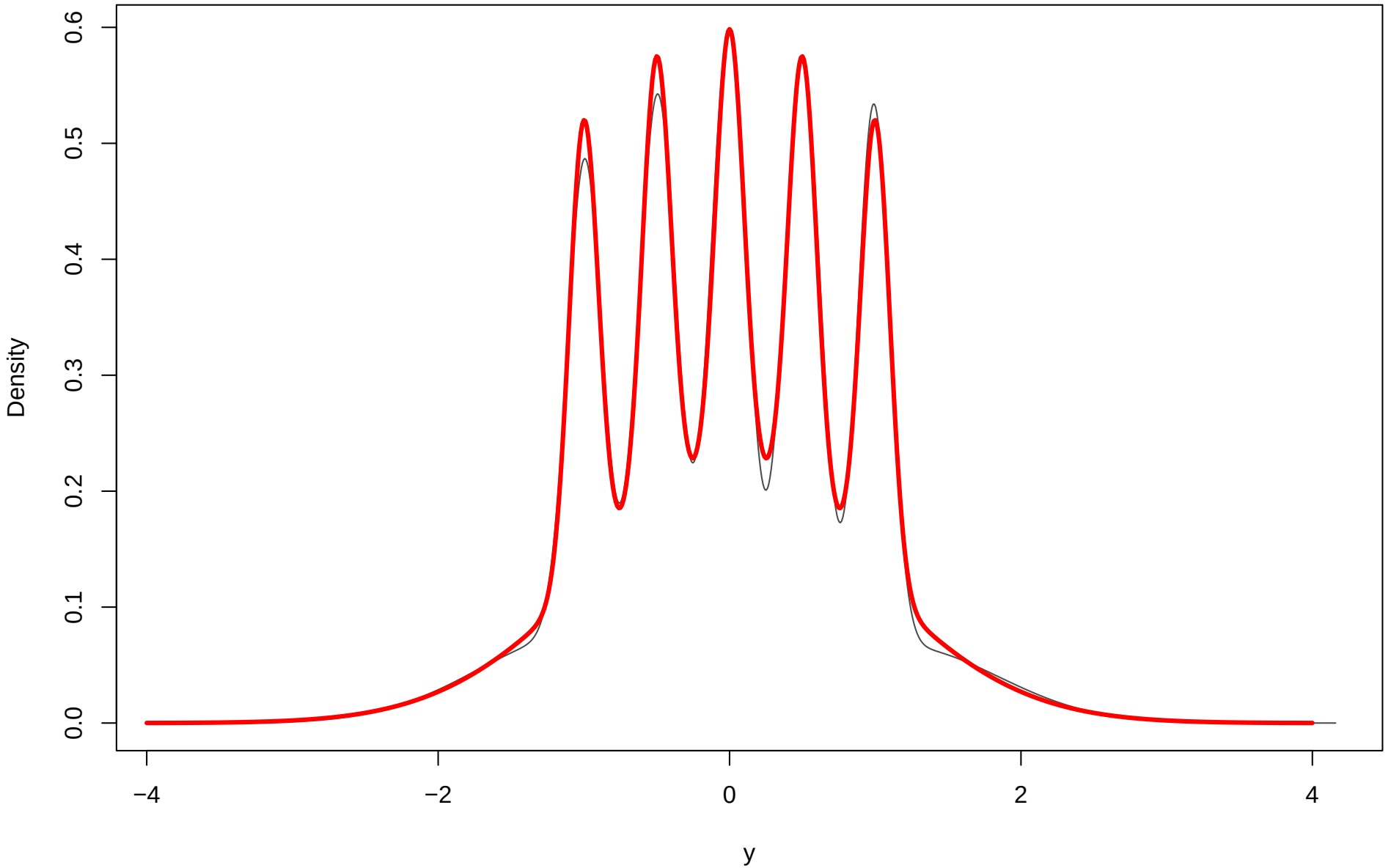


```
plot(mod, what = "classification", main = FALSE)
```



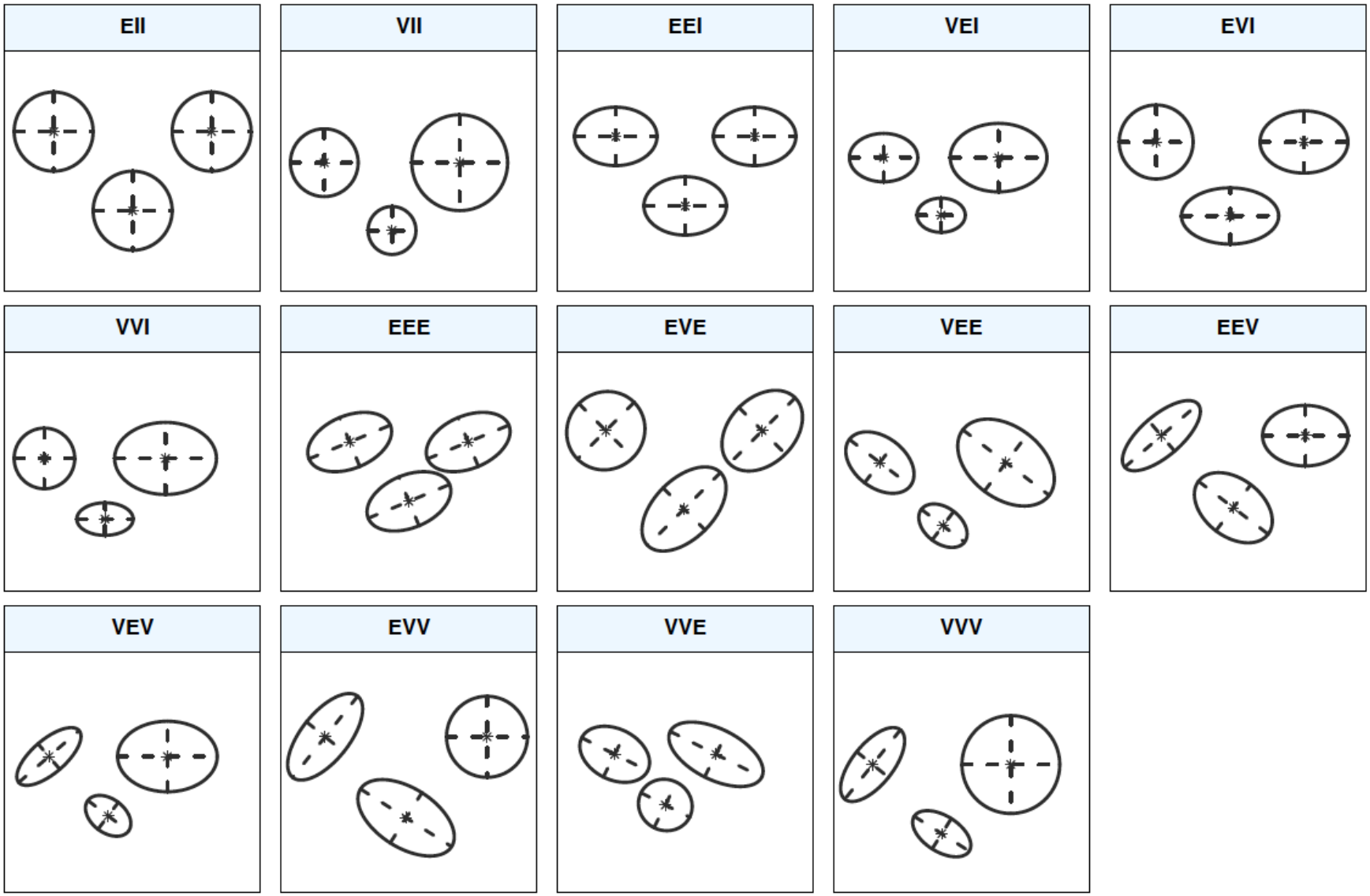
```
plot(mod, what = "density", main = FALSE)
```




Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
−4
−2
0
2
4
y

Extending the Model - multivariate

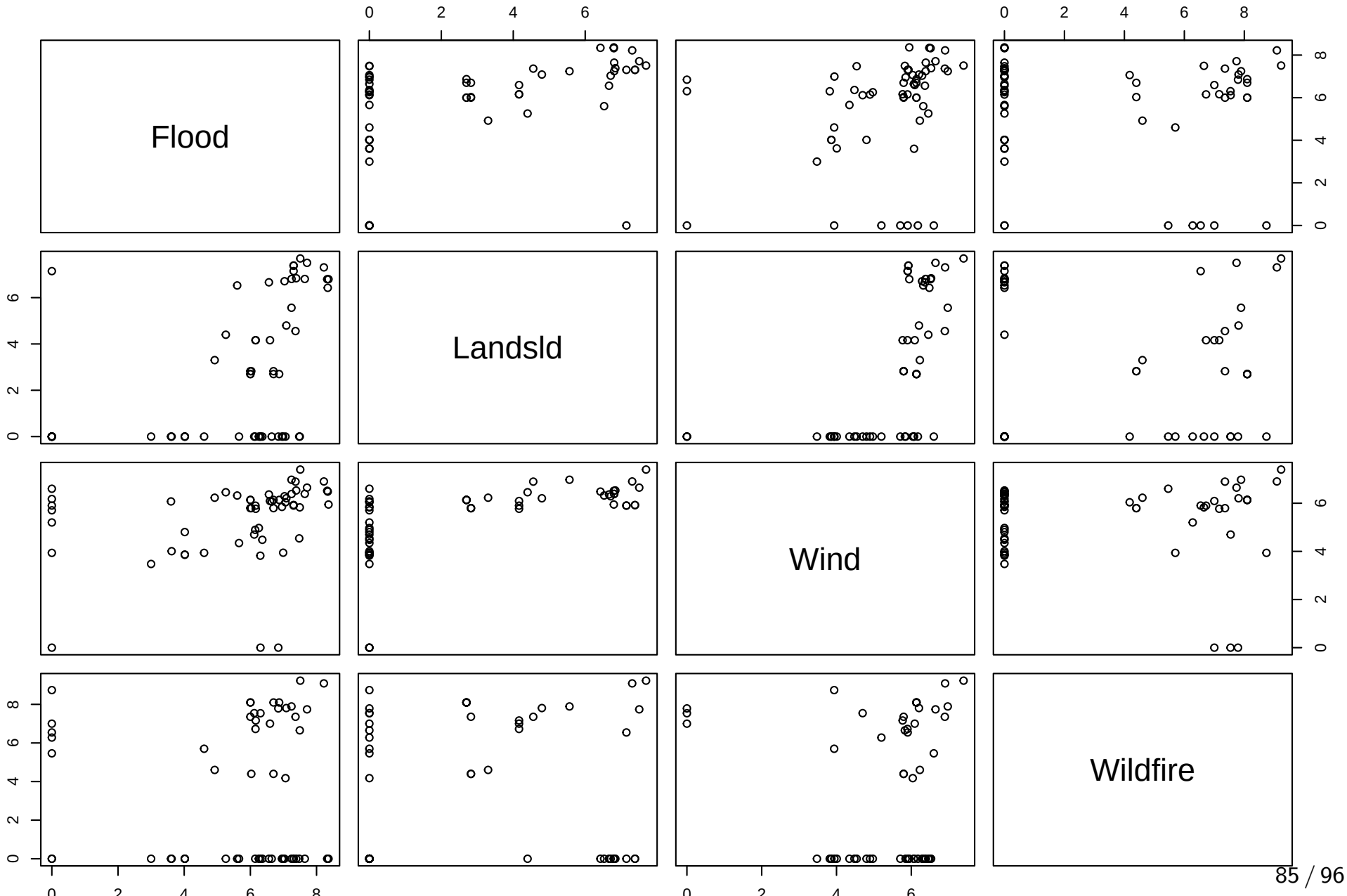
Model	$\boldsymbol{\Sigma}_k$	Distribution	Volume	Shape	Orientation
EII	$\lambda \boldsymbol{I}$	Spherical	Equal	Equal	—
VII	$\lambda_k \boldsymbol{I}$	Spherical	Variable	Equal	—
EEI	$\lambda \boldsymbol{A}$	Diagonal	Equal	Equal	Coordinate axes
VEI	$\lambda_k \boldsymbol{A}$	Diagonal	Variable	Equal	Coordinate axes
EVI	$\lambda \boldsymbol{A}_k$	Diagonal	Equal	Variable	Coordinate axes
VVI	$\lambda_k \boldsymbol{A}_k$	Diagonal	Variable	Variable	Coordinate axes
EEE	$\lambda \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^\top$	Ellipsoidal	Equal	Equal	Equal
EVE	$\lambda \boldsymbol{D}\boldsymbol{A}_k\boldsymbol{D}^\top$	Ellipsoidal	Equal	Variable	Equal
VEE	$\lambda_k \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^\top$	Ellipsoidal	Variable	Equal	Equal
VVE	$\lambda_k \boldsymbol{D}\boldsymbol{A}_k\boldsymbol{D}^\top$	Ellipsoidal	Variable	Variable	Equal
EEV	$\lambda \boldsymbol{D}_k\boldsymbol{A}\boldsymbol{D}_k^\top$	Ellipsoidal	Equal	Equal	Variable
VEV	$\lambda_k \boldsymbol{D}_k\boldsymbol{A}\boldsymbol{D}_k^\top$	Ellipsoidal	Variable	Equal	Variable
EVV	$\lambda \boldsymbol{D}_k\boldsymbol{A}_k\boldsymbol{D}_k^\top$	Ellipsoidal	Equal	Variable	Variable
VVV	$\lambda_k \boldsymbol{D}_k\boldsymbol{A}_k\boldsymbol{D}_k^\top$	Ellipsoidal	Variable	Variable	Variable


EII
VII
EEI
VEI
EVI
VVI
EEE
EVE
VEE
EEV
VEV
EVV
VVE
VVV

California

```
Cali <- read.csv("Cali.csv", header=TRUE)
cali <- Cali[, 2:5]
pairs(cali)
```

California



```r
mod <- Mclust(cali)
tab <- summary(mod, parameters = TRUE)
summary(mod)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 8 components:
##
##  log-likelihood  n df       BIC       ICL
##       -176.4069 58 98 -750.7372 -750.9149
##
## Clustering table:
##  1  2  3  4  5  6  7  8
## 12 10  5  6  7 12  3  3
```

```
tab$pro
```

```
##          1          2          3          4          5          6          7
## 0.20689650 0.17094554 0.08620628 0.10344828 0.12071320 0.20834212 0.05172394
##          8
## 0.05172414
```

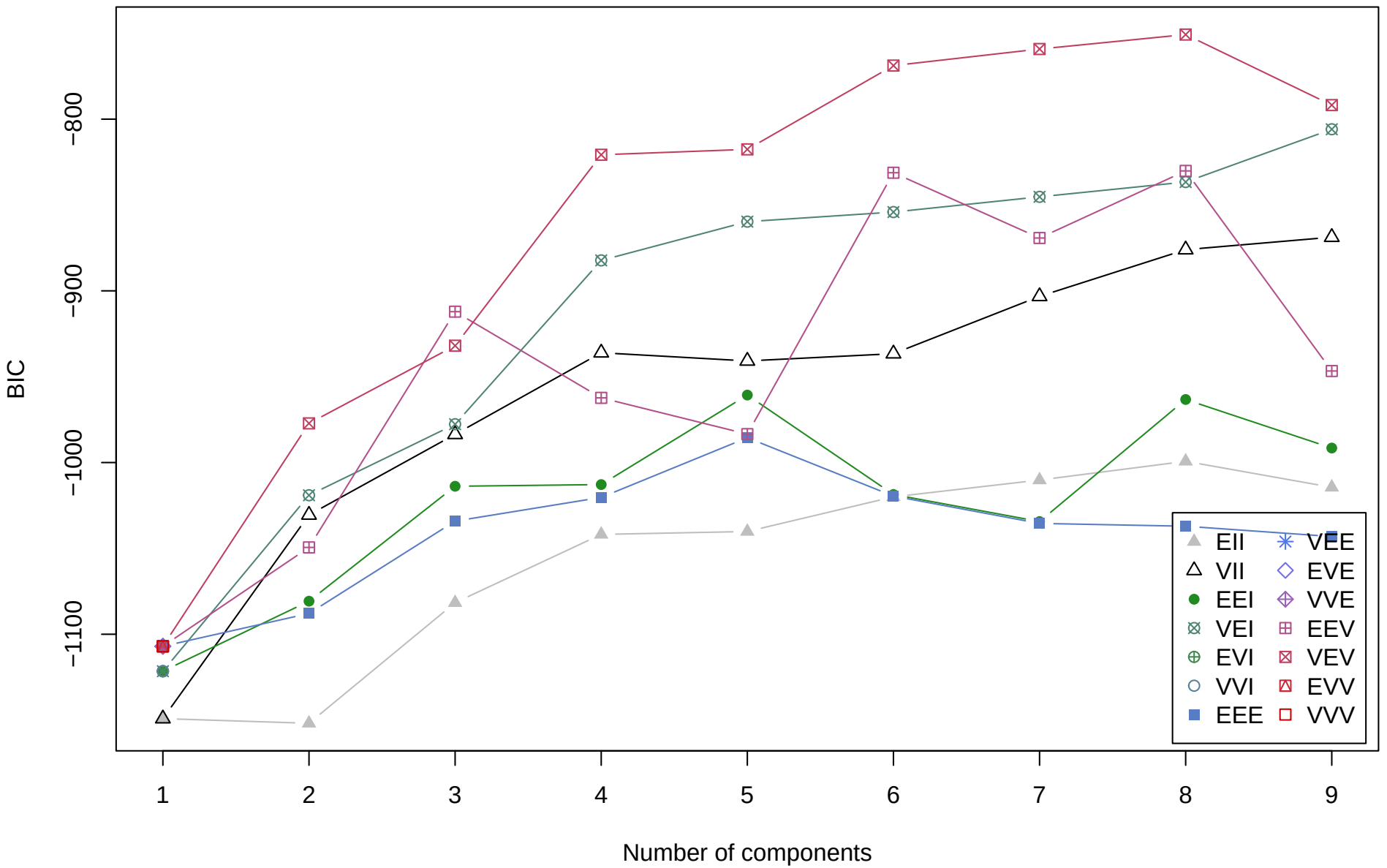
```
tab$mean
```

```
##              [,1]     [,2]     [,3]          [,4]     [,5]     [,6]     [,7]
## Flood    7.367987 6.353397 3.734891 1.086194e-258 6.455926 6.170086 7.814561
## Landsld  6.856285 3.200957 0.000000  1.190892e+00 0.000000 1.872688 7.500267
## Wind     6.247657 6.015101 4.002401  5.587727e+00 4.428409 5.901164 6.983867
## Wildfire 0.000000 7.444942 0.000000  4.504000e+00 0.000000 3.917816 8.684974
##                  [,8]
## Flood     4.383483e+00
## Landsld   0.000000e+00
## Wind     1.182133e-105
## Wildfire  7.443380e+00
```

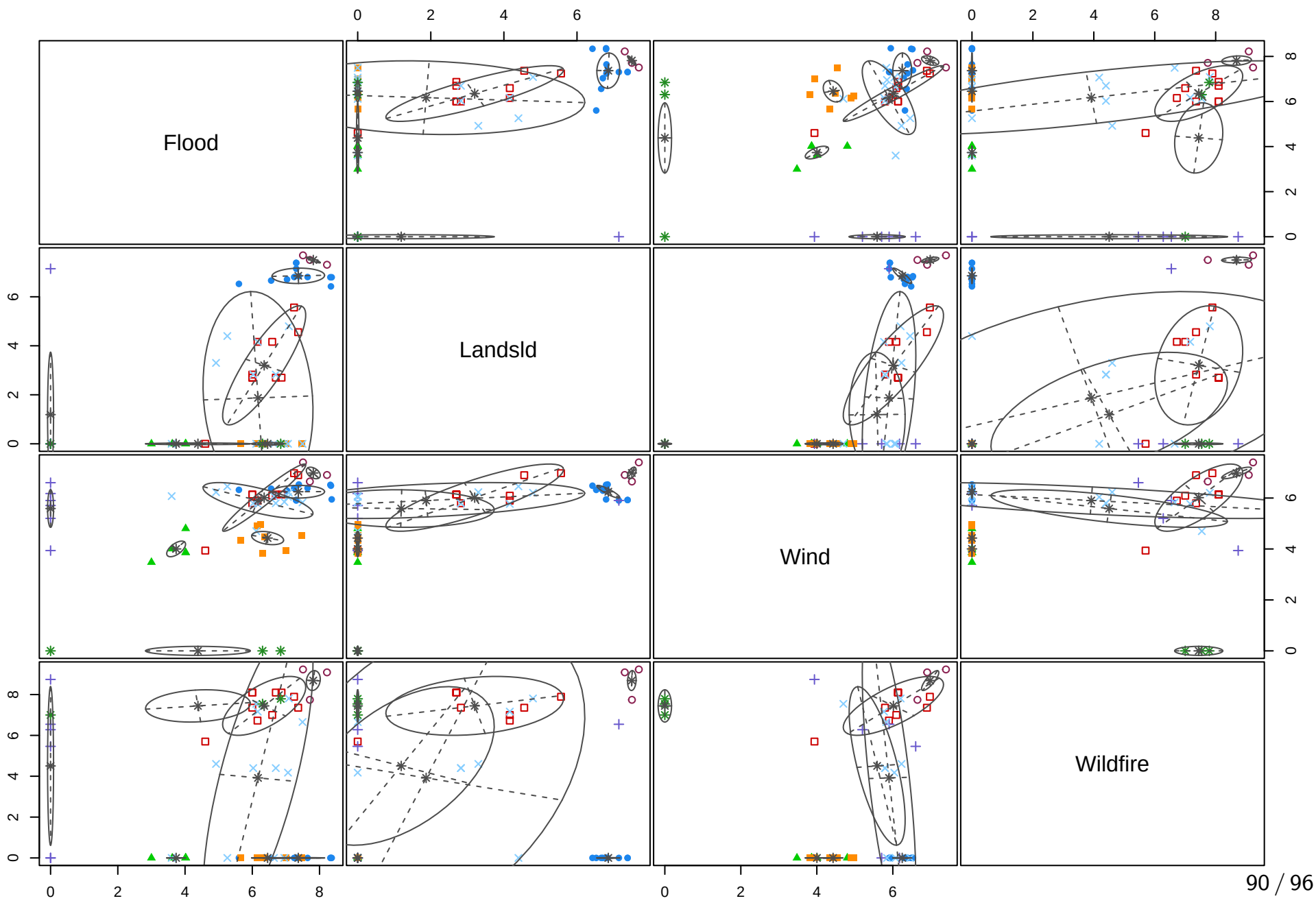
```
tab$variance[,,1]
```

```
##                Flood     Landsld         Wind     Wildfire
## Flood    0.612010567  0.01427758  0.003768758 0.0000000000
## Landsld  0.014277577  0.09798743 -0.069535035 0.0000000000
## Wind     0.003768758 -0.06953504  0.061749448 0.0000000000
## Wildfire 0.000000000  0.00000000  0.000000000 0.0002754592
```

```
plot(mod, what = "BIC", main = FALSE)
```



```
plot(mod, what = "classification", main = FALSE)
```



Let's Examine the Groups

```
prob <- mod$z
head(prob)

max <- NULL
for(i in 1:nrow(prob)){
  max.i <- which.max(prob[i,])
  max <- c(max, max.i)
}
```

Let's Examine the Groups

```
##      [,1]          [,2]         [,3] [,4]        [,5]          [,6] [,7]
## [1,]    1 3.143739e-202 0.000000e+00    0 0.00000e+00 5.683134e-10    0
## [2,]    0  9.995440e-01 0.000000e+00    0 0.00000e+00 4.560439e-04    0
## [3,]    0 2.301819e-245 1.000000e+00    0 1.28986e-09 2.119720e-35    0
## [4,]    0 1.790323e-245 0.000000e+00    1 0.00000e+00 8.461551e-14    0
## [5,]    0 2.301819e-245 1.000000e+00    0 1.28986e-09 2.119720e-35    0
## [6,]    0  0.000000e+00 1.822878e-17    0 1.00000e+00 4.510380e-14    0
##                [,8]
## [1,]  0.000000e+00
## [2,]  0.000000e+00
## [3,] 1.246040e-126
## [4,] 5.215790e-190
## [5,] 1.246040e-126
## [6,] 1.489638e-190
```

Group 1

```
as.character(Cali[,1][max==1])
```

```
##  [1] "Alameda"     "ContraCosta" "DelNorte"    "Humboldt"    "Marin"
##  [6] "Mendocino"   "Monterey"    "Napa"        "SanMateo"    "SantaClara
## [11] "SantaCruz"   "Sonoma"
```

Group 2

```
as.character(Cali[,1][max==2])
```

```
##  [1] "Alpine"    "ElDorado"  "Fresno"    "Kern"      "Madera"    "Nevada
##  [7] "Placer"    "Plumas"    "Riverside" "Shasta"
```

Group 3

```
as.character(Cali[,1][max==3])
```

```
## [1] "Amador"    "Calaveras" "Imperial"  "Tuolumne"  "Yolo"
```

Some Good Sources

- *Finite Mixture Models* - G. McLachlan and D. Peel
- *The Elements of Statistical Learning* - T. Hastie, R. Tibshirani, J. Friedman
- "mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models" (on Wattle) - L. Scrucca, M. Fop, T. Murphy, and A. Raftery