Data Wrangling and Data Analysis
Unsupervised learning: Model-based clustering

Daniel Oberski & Erik-Jan van Kesteren

Department of Methodology & Statistics

Utrecht University

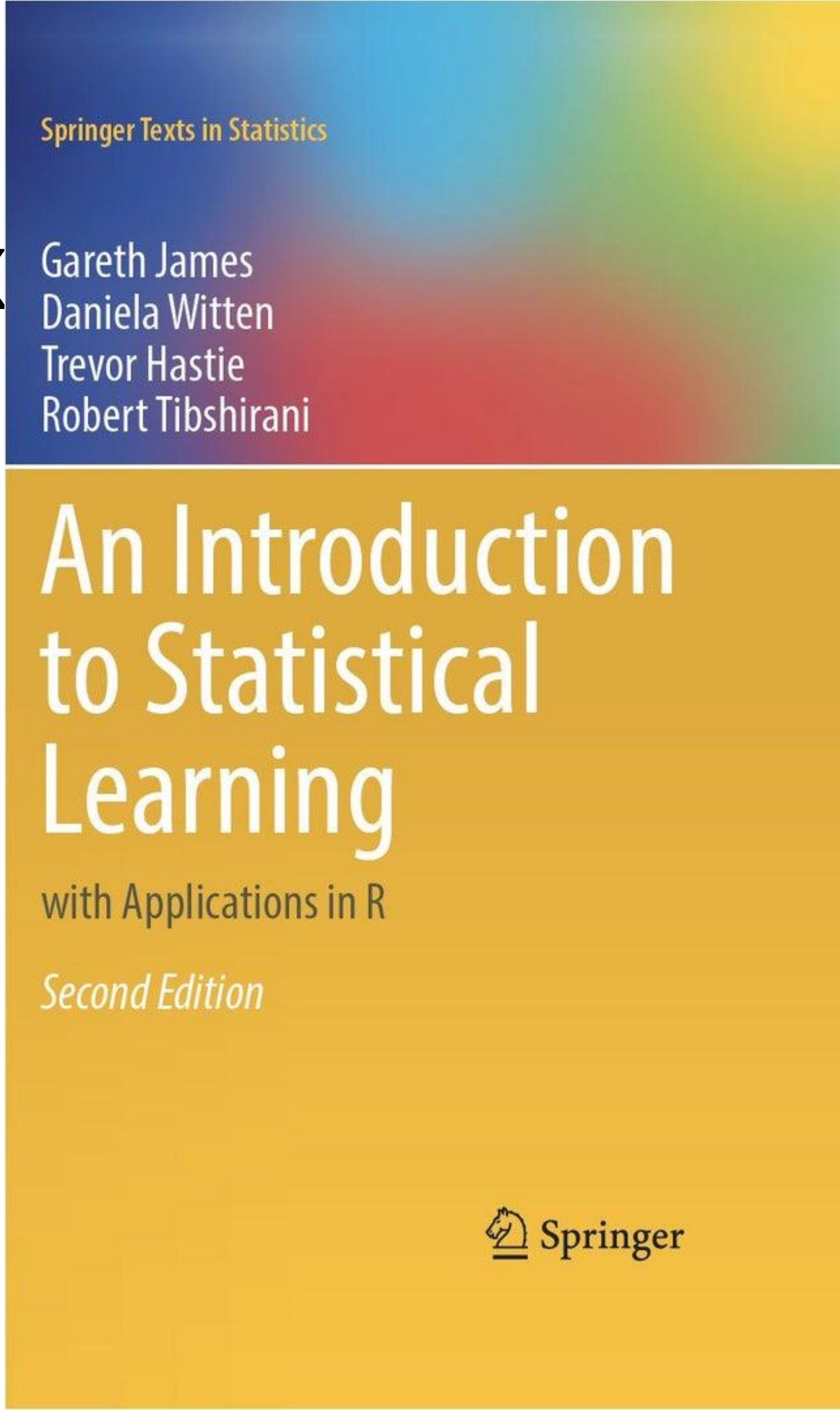
This week

- **Lecture 1: Clustering #2: Model-based clustering**
- Lecture 2: Text mining #1
- Lecture 3: Text mining #2

Reading materials about clustering (this week & nex

- Selected paragraphs from **Introduction to Statistical Learning (ISLR)** §12.1 and 12.4

- "Mixture models: latent profile and latent class analysis" [Oberski, 2016] §1, §2
  - http://daob.nl/wp-content/papercite-data/pdf/oberski2016mixturemodels.pdf



Optional, much more in-depth material

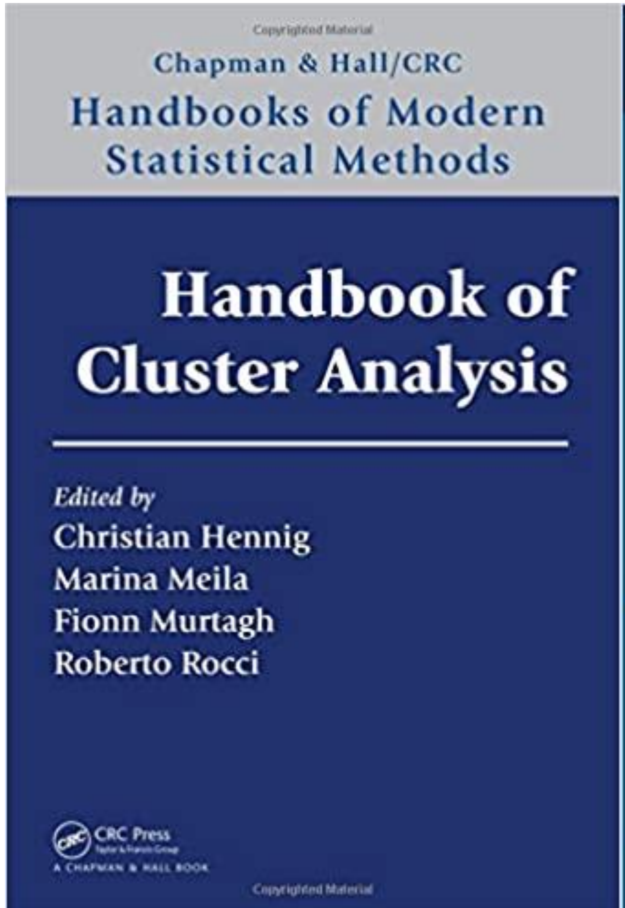
Clustering strategy and method selection (ch. 31), https://arxiv.org/pdf/1503.02059.pdf

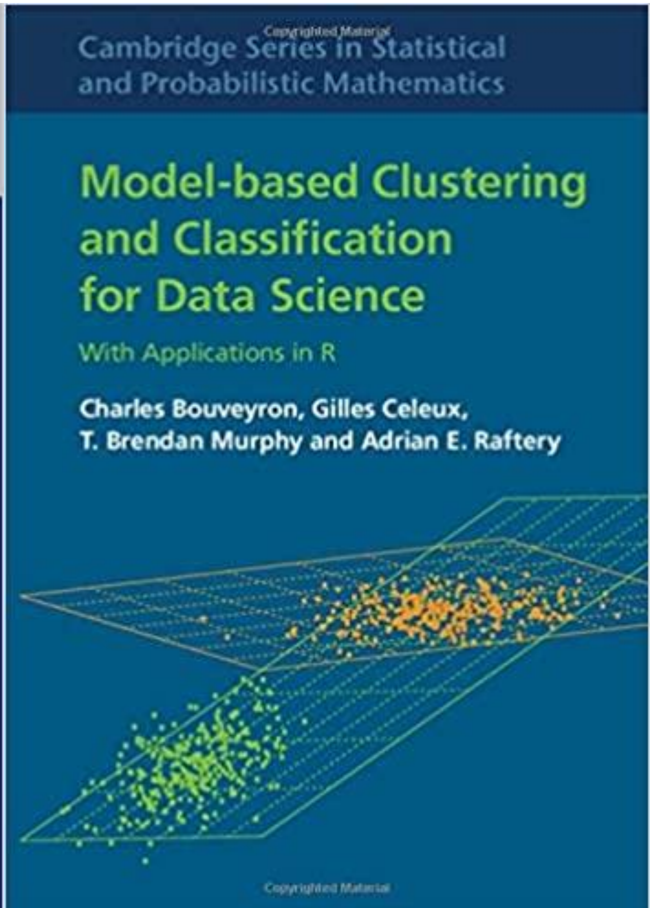
Handbook of Cluster Analysis

Hennig et al. (2016)

Model-based Clustering and Classification for Data Science

Bouveyron et al. (2018)





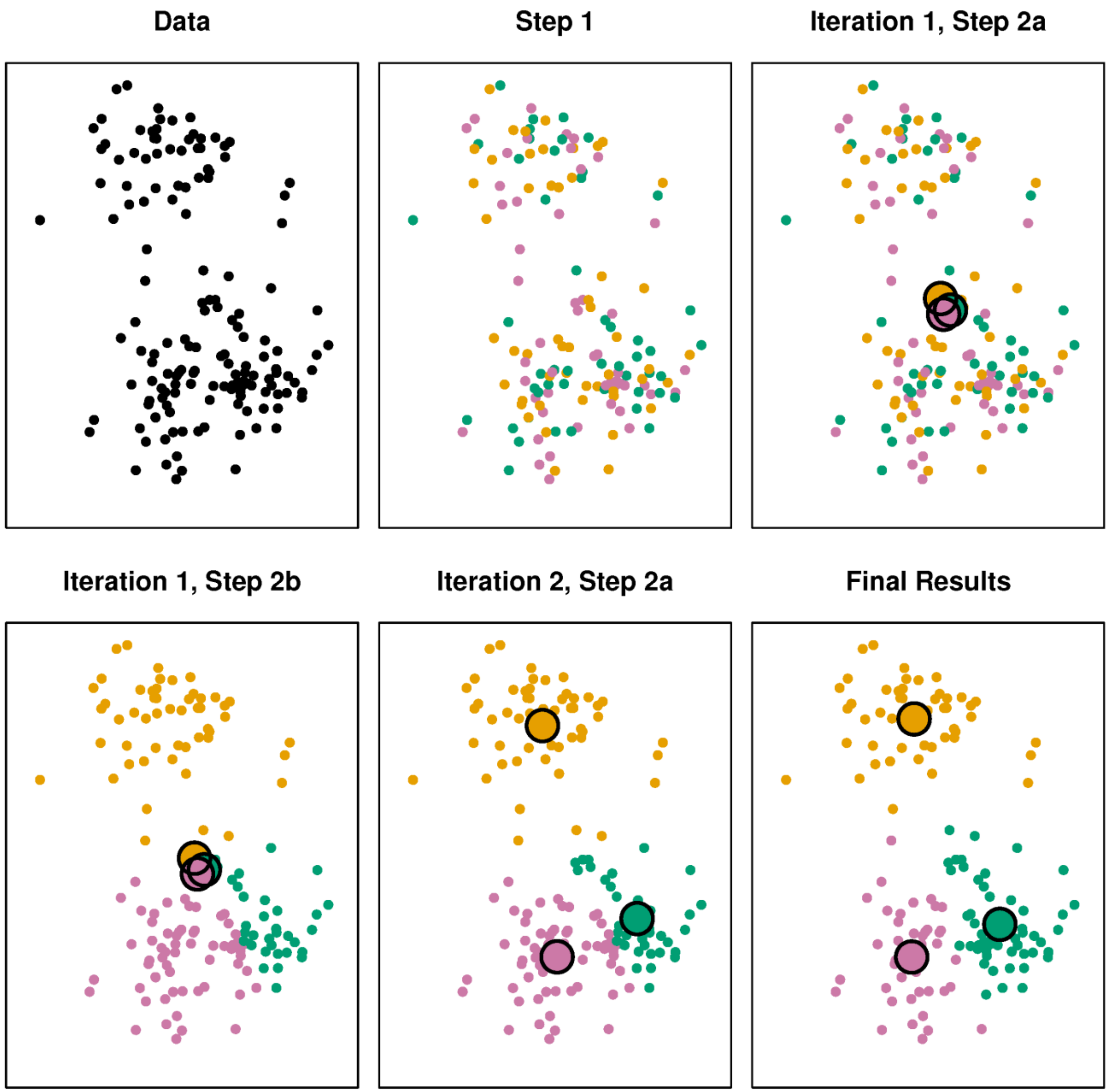
Model-based clustering

K-means again

1. Assign examples to K clusters

2.
    a. Calculate K cluster centroids;
    b. Assign examples to cluster with closest centroid;

3. If assignments changed, back to step 2a; else stop.



K-means again

- K-means is based on a **rule**
- Why this rule and not some other rule?
- What kind of data does the rule work well for?
- In what situations would the rule fail?
- What happens if we want to change the rule?

All **difficult to answer by staring at the algorithm.**

Model-based clustering

Steps:

1. Pretend we believe in some *statistical model* that describes data as belonging to unobserved (“latent”) groups;
2. Estimate (“train”) this model using the data.

- **The rule follows from the model!**
- Instead of worrying about *algorithm*, we worry about *model*.
- Questions are easy to answer.

Model-based clustering

- Assumptions about the clusters are explicit, not implicit.
- We will look at the most commonly used family of models,

Gaussian mixture models (GMMs):

- Data within each cluster *(multivariate) normally distributed.*
- Parameters can be either the same or different across groups:
  - Volume (size of the clusters in data space);
  - Shape (circle or ellipse);
  - Orientation (the angle of the ellipse).

Model-based clustering

Another major advantage:

- For each observation, get a **posterior probability** of belonging to each cluster;
- Reflects that cluster membership is uncertain;
- Cluster assignment can be done based on the highest probability cluster for each observation.

Model-based clustering

Specific examples of model-based clustering:

- Gaussian mixture models
- Latent profile analysis
- Latent class analysis (categorical observations)
- Latent Dirichlet allocation

Gaussian mixture modeling

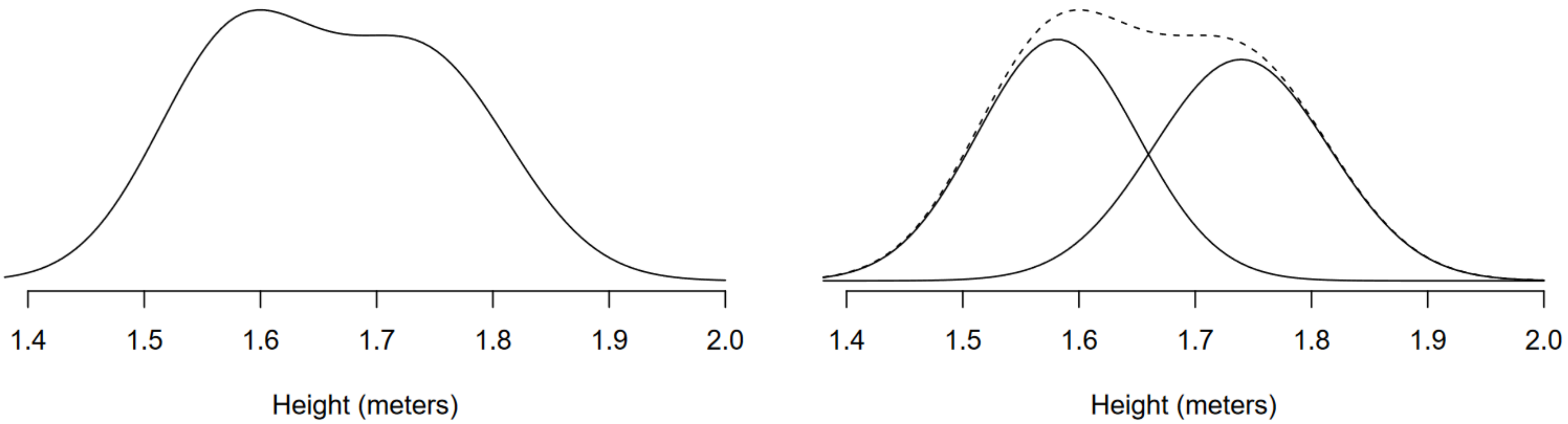


Fig. 1 Peoples' height. Left: observed distribution. Right: men and women separate, with the total shown as a dotted line.

Model-based clustering

- Statistical model + assumptions defines a likelihood

$$p(\text{data} \mid \text{parameters}) = p(y \mid \theta)$$

- Maximum likelihood estimation: find the parameters θ that make it most likely to observe the data we actually observed, y

- The above procedure automatically gives algorithm for computing clusters from data, given the model.

Model-based clustering

Likelihood (*density*) for height data:

$$p(\text{height} \mid \theta) =$$

$$\Pr(\text{man}) \cdot \text{Normal}(\mu_{\text{man}}, \sigma_{\text{man}}) +$$

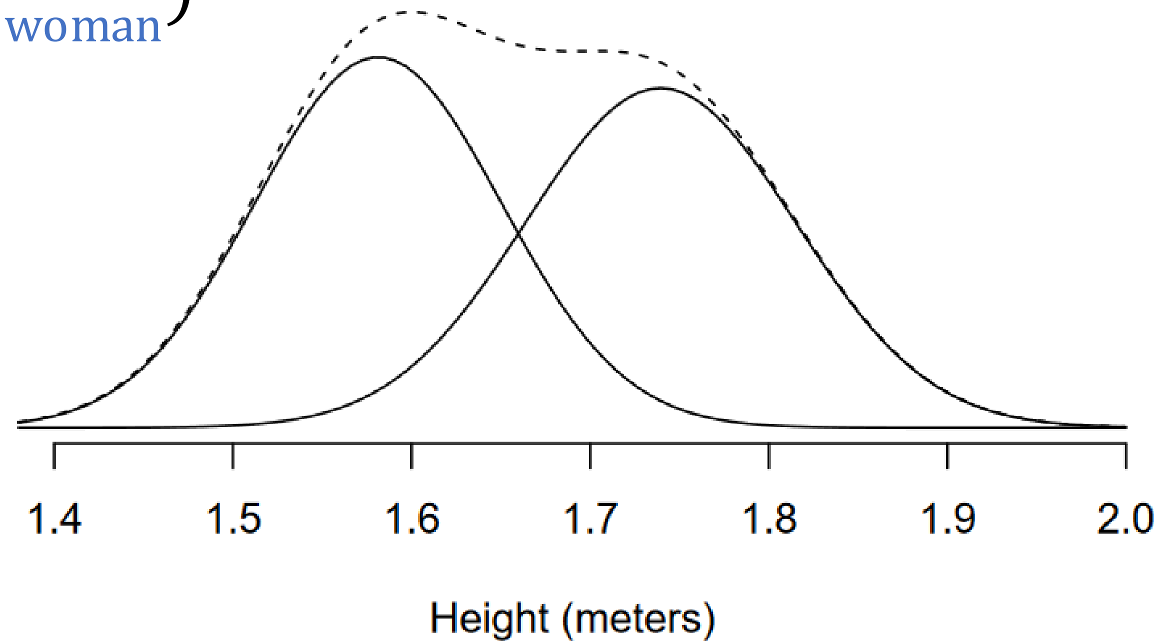
$$\Pr(\text{woman}) \cdot \text{Normal}(\mu_{\text{woman}}, \sigma_{\text{woman}})$$

Or, more concise notation:

$$p(\text{height} \mid \theta) =$$

$$\pi_1^X \text{Normal}(\mu_1, \sigma_1) +$$

$$(1 - \pi_1^X) \text{Normal}(\mu_2, \sigma_2)$$



Model-based clustering

Gaussian mixture model **parameters**:

- π_1^X determines the relative cluster sizes
  - Proportion of observations to be expected in each cluster
- μ_1 and μ_2 determine the locations of the clusters
  - Like centroids in K-means clustering
- σ_1 and σ_2 determine the volume of the clusters
  - how large / spread out the are clusters are in data space

Together, these 5 unknown parameters describe our model of how the data is generated.

Estimation: the EM algorithm

- If we knew in advance who is a man and who is a woman, it would have been easy to find the estimates for μ and σ:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{N_1} \mathtt{height}_i}{N_1}, \qquad \hat{\sigma}_1 = \sqrt{\frac{\sum_{i=1}^{N_1} (\mathtt{height}_i - \hat{\mu}_1)^2}{N_1}}$$

(and same for $\hat{\mu}_2$ and $\hat{\sigma}_2$.)

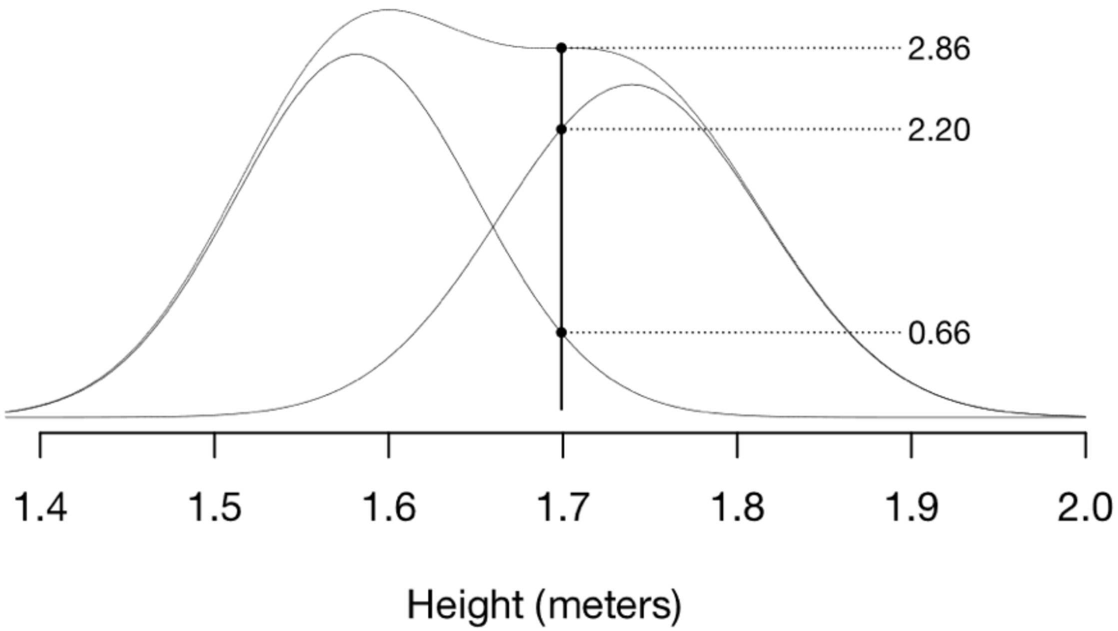
- But we don't know this!

-> Assignments need to be estimated too.

Estimation: the EM algorithm

- Solution: Figure out the posterior probability of being a man/woman, given the current estimates of the means and sds
- If we know cluster locations and shapes, how likely is it that a 1.7m person is a man or a woman?

$$\pi^{X}_{man} = \frac{2.20}{2.86} \approx 0.77$$

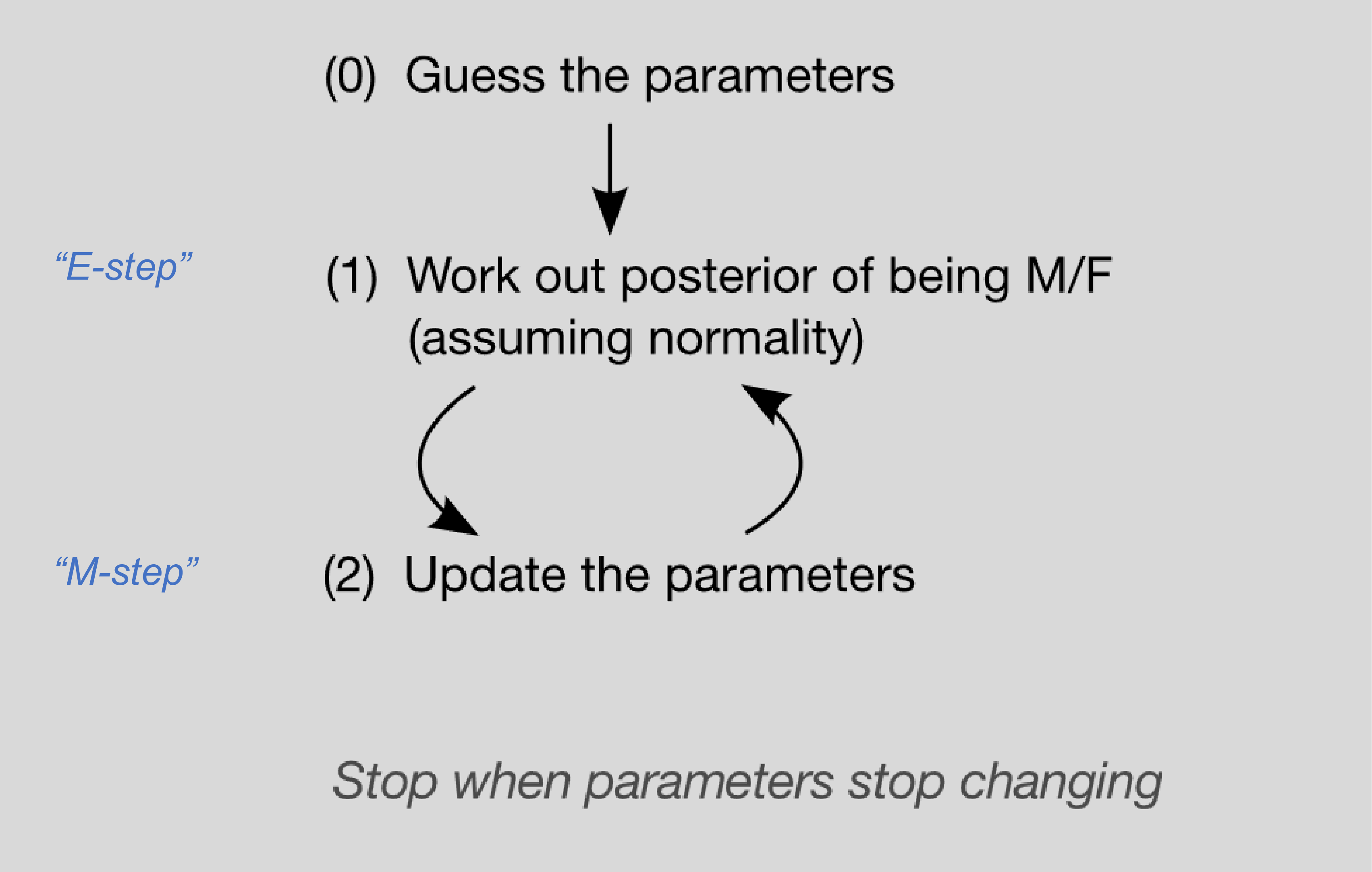


Estimation: the EM algorithm

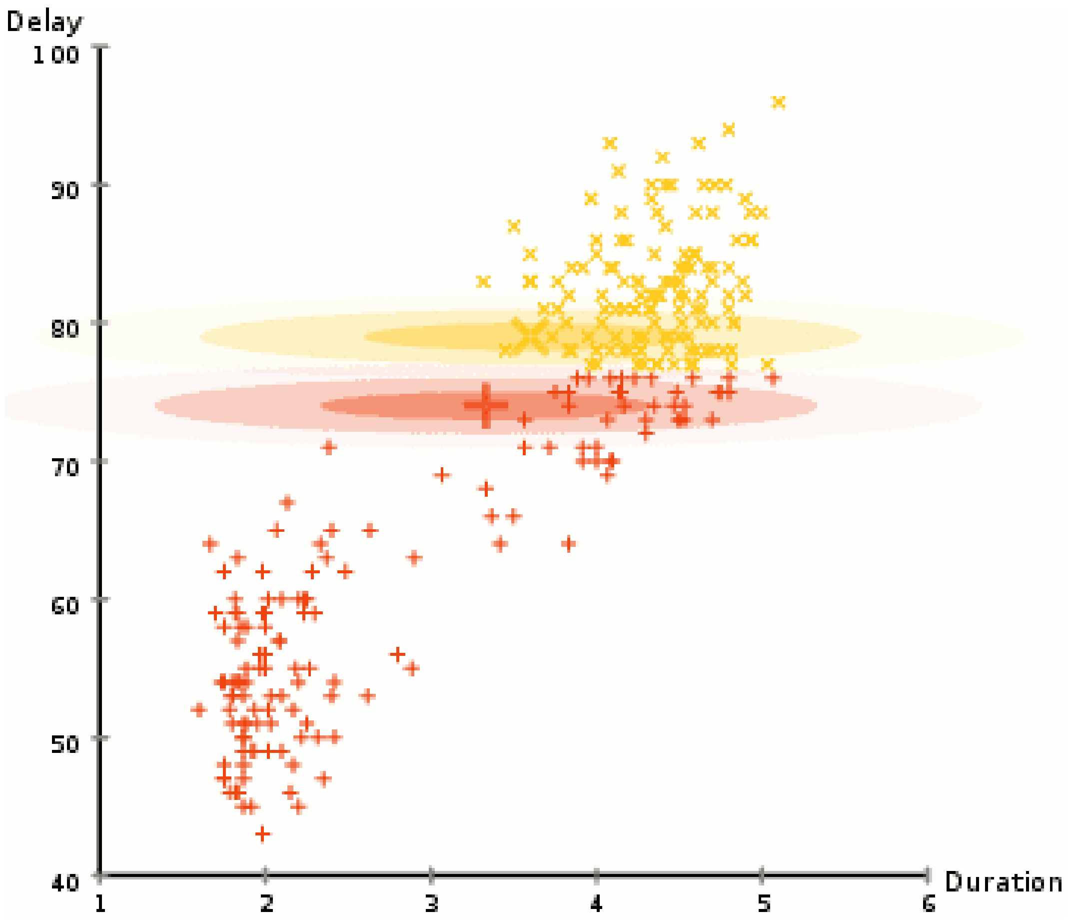
- Now we have some class *assignments* (probabilities);
- So we can go back to the parameters and update them using our easy rule (M-step)
- Then, we can compute new posterior probabilities (E-step)

Does it remind you of something…?

Estimation: the EM algorithm



Estimation: the EM algorithm

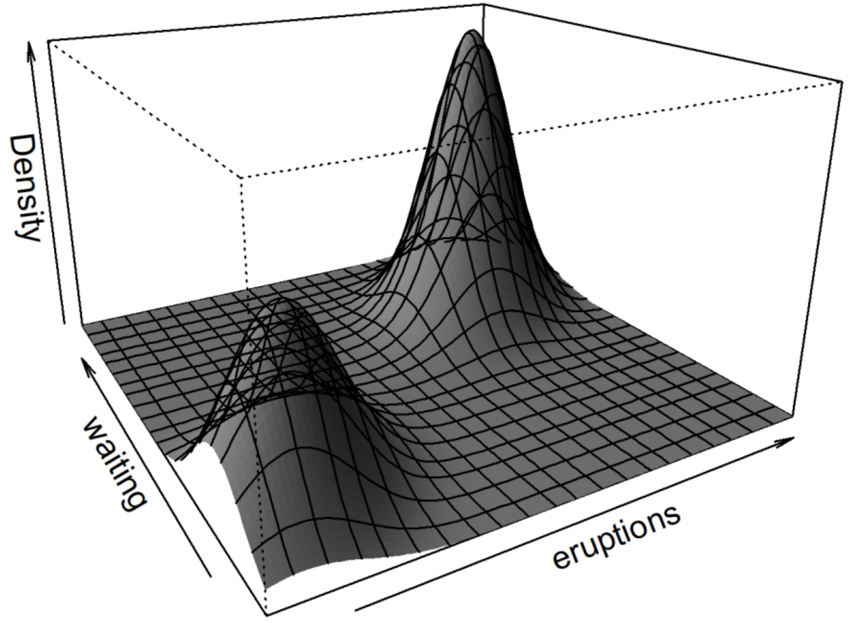


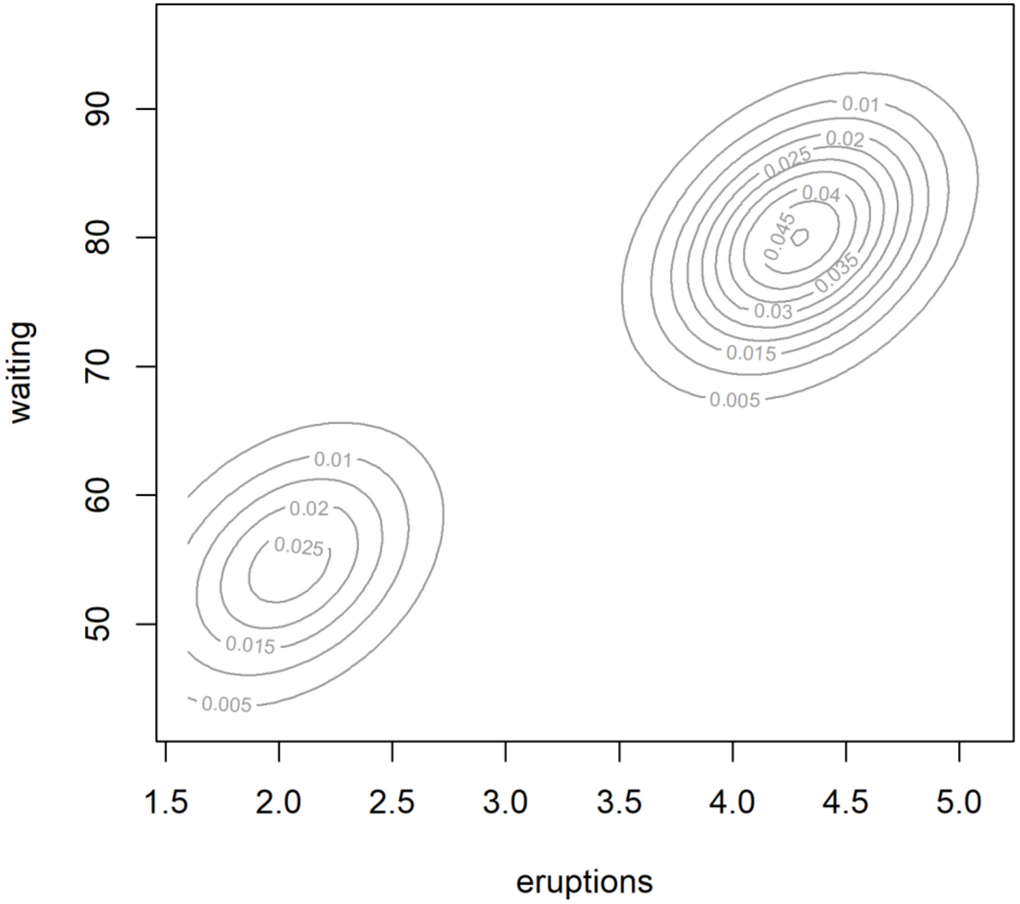
Multivariate model-based clustering

- With 2 observed features:
  - mean becomes a vector of 2 means
  - standard deviation turns into a 2x2 variance-covariance matrix determining the shape of the cluster
- So we have multiple within-cluster parameters:
  - Two means
  - Two variances, one for each observed variable
  - A single covariance among the features
- Together, the 11 parameters define the likelihood in bivariate space, which from the top looks like ellipses

Multivariate model-based clustering

$$p(\boldsymbol{y}|\,\theta) = \pi_1^X MVN(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}) + (1\ - \pi_1^X)MVN(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2})$$





Number of parameters in a (multivariate) Gaussian mixture model

The number of parameters in a multivariate mixture model is:

- (the π_k^X) The number of components (classes), minus one, i.e. $K - 1$
- (the $\boldsymbol{\mu}_k$), i.e. $\mathrm{K} \cdot p$ (where p is the number of variables)
- (the $\boldsymbol{\Sigma}_k$), i.e.
    - $K \cdot p$ variances,
        - (or p variances when variances **equal over classes**)
    - $K \cdot p\ (p-1)/2$ covariances
        - (or $p\ (p-1)/2$ when covariances **equal over classes**)
        - (or 0 when variables are uncorrelated, spherical clusters)

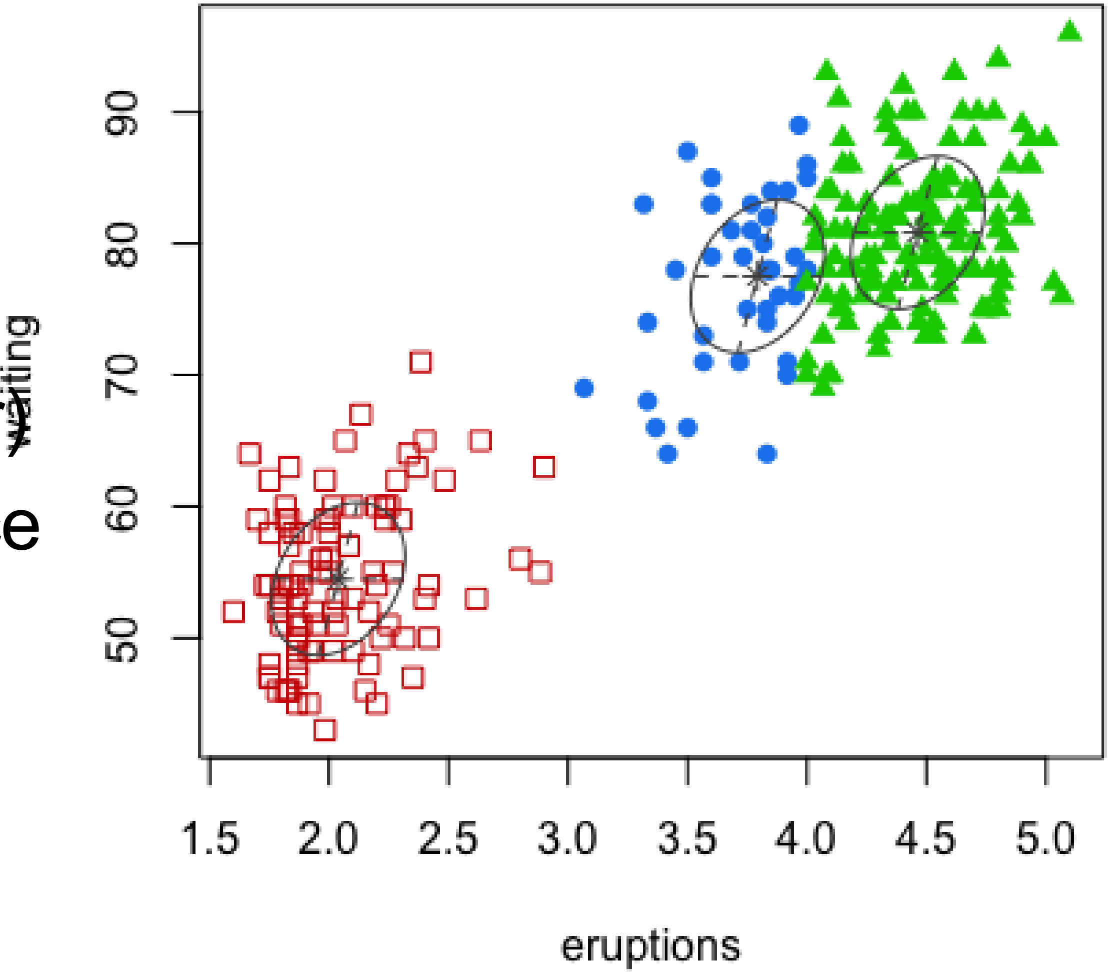
Number of parameters

$$m = (K - 1) + Kp + Kp + K\frac{p(p-1)}{2}$$

For example:

- $K = 3$
- $p = 2$
- Ellipsoidal (correlated within cluster)
- But: equal variances and covariance

$$m = (K - 1) + Kp + p + \frac{p(p-1)}{2}$$
$$= 2 + 3 \times 2 + 2 + 1$$
$$= 11$$



Multivariate model-based clustering

- Cluster shape parameters (the variance-covariance matrix) *can* be constrained to be *equal* across clusters
- Can also be *different* across clusters
- More flexible, complex model
- Think: **bias-variance tradeoff**

How to evaluate clustering results

1. Use of external information
2. Visual exploration
3. Stability assessment / sensitivity analysis
4. Internal validation indexes
5. **Testing for clustering structure**

Much more info & helpful advice: Clustering strategy & method selection (ch 31 of Handbook of clustering), https://arxiv.org/pdf/1503.02059.pdf

Model fit

- The likelihood says how well the model fits to the data
- It forms the basis of information criteria (lower is better)
  - Can be used to compare different clustering models and pick the best one

$$BIC = -2 \cdot \log(\ell) + m \cdot \log(n)$$

- ℓ : Likelihood, $p(\text{data} \mid \theta)$
- $-2 \cdot \log(\ell)$: *"Deviance"*
- m : Number of parameters
- n : Number of observations/examples

Model fit

- Tradeoff between fit and complexity

$$\underbrace{-2 \cdot \log(\ell)}_{\text{“Training data fit”}} + \underbrace{m \cdot \log(n)}_{\text{“Model capacity”}}$$

- Think: bias and variance tradeoff
  - Variance also has to do with “clustering stability”
- Better fit *and* lower complexity = better cluster solution

More model fit criteria

- BIC: “Schwarz/Bayesian information criterion”
- AIC: “Another/Akaike information criterion”

  (same as BIC but penalty is m)
- AIC3: The same as AIC but penalty is $\frac{3}{2}m$
- ICL: “Integrated information criterion” (Biernacki et al. 2000)

(Same as BIC but reconstruction loss includes the assigned clusters)

- *(Others based on):*
  - *Minimum description length (MDL)*
  - *Bayesian marginal likelihood*

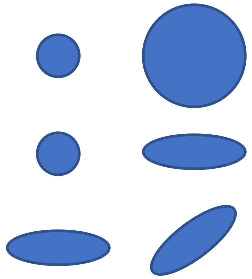
Model-based clustering in R/Python

- `mclust` implements multivariate model-based clustering
- Provides an easy interface to fit several parameterizations
- Model comparison with BIC
- Plotting functionality
- In Python: `sklearn.mixture.GaussianMixture`

```
> library(mclust)
    __  ___________    __  _____________
   /  |/  / ____/ /   / / / / ___/_  __/
  / /|_/ / /   / /   / / / /\__ \ / /
 / /  / / /___/ /___/ /_/ /___/ // /
/_/  /_/\____/_____/\____//____//_/    version 5.4.6
```

Model-based clustering in R

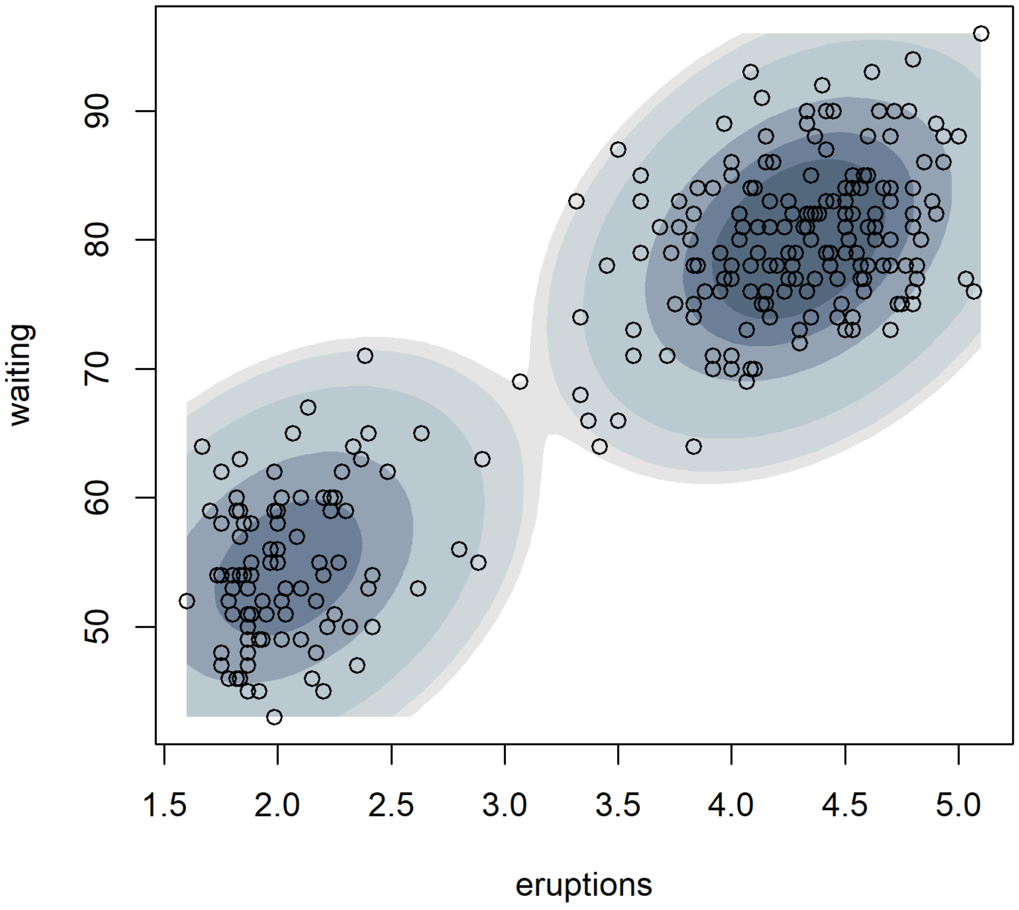
- Mclust uses an identifier for each possible parametrization :
- **E** for **e**qual, **V** for **v**ariable, **I** for identity matrix:
  - Volume (size of the clusters in data space):
  - Shape (circle or ellipse)
  - Orientation (the angle of the ellipse)



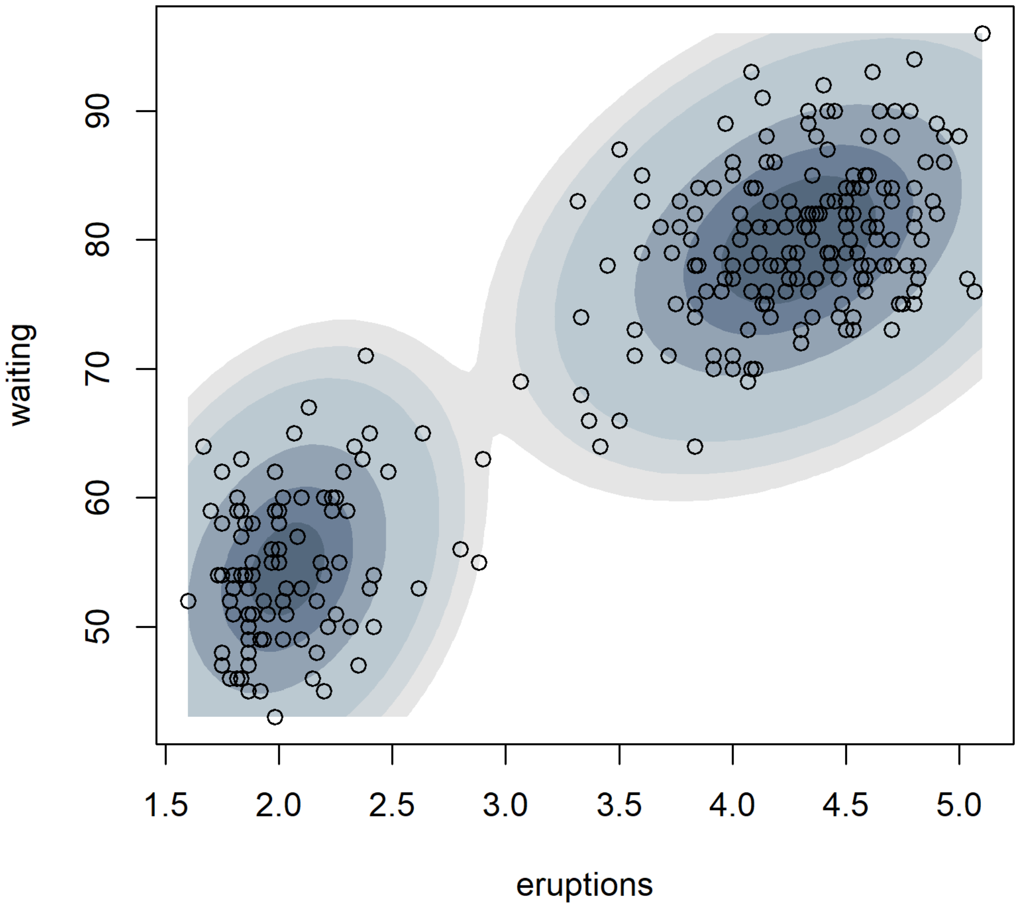
- E.g. an “EEE” model has equal volume, shape and orientation
- A VVV model has variable volume, shape, and orientation
- A VVE model has variable volume and shape but equal orientation

Model-based clustering in R:
EEE vs. VVV

Equal volume, shape, orientation

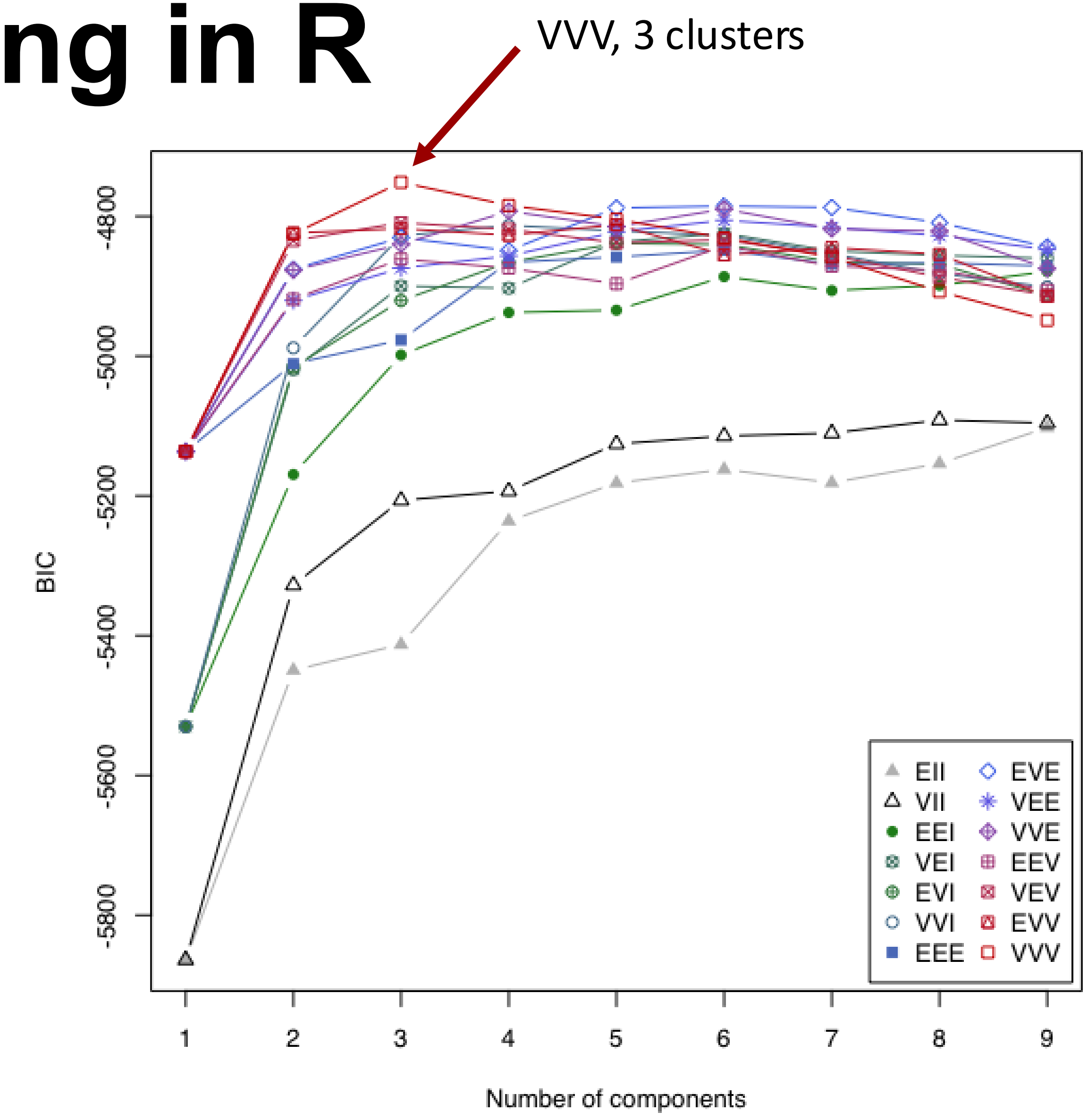


Variable volume, shape, orientation



Model-based clustering in R

- How `mclust` optimizes hyperparameters:
  - Fit all the models with up to 9 clusters (or more, your choice!)
  - Compute the BIC (or ICL) of each model
  - Choose the model with the best BIC
- R assignment: using mclust



Merging *components* to get *clusters*

- GMM obviously has trouble with clusters that are not ellipses
- Secret weapon: **merging**

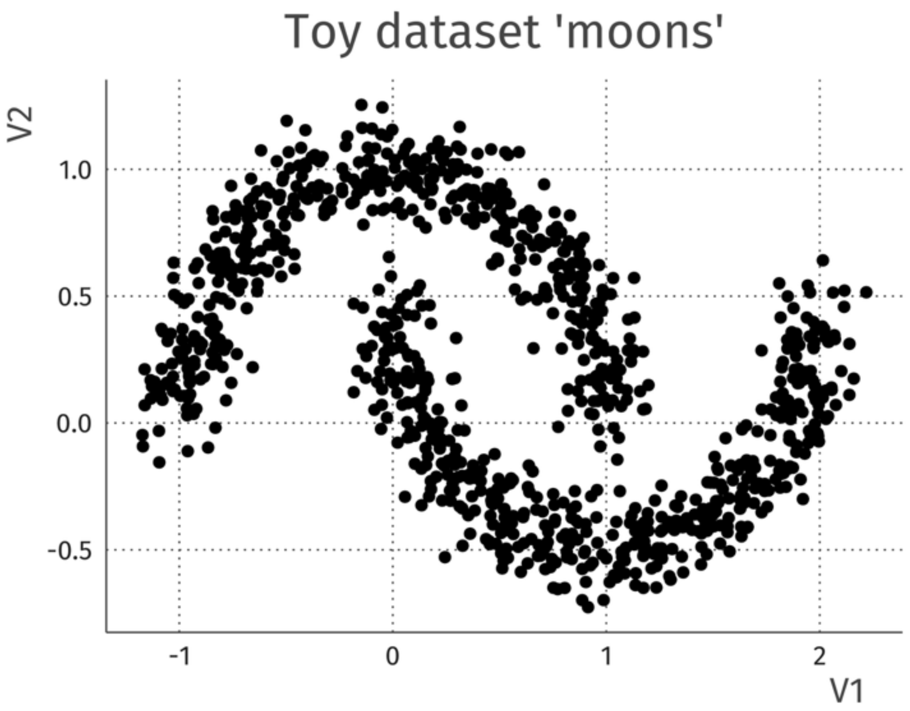
Powerful idea:

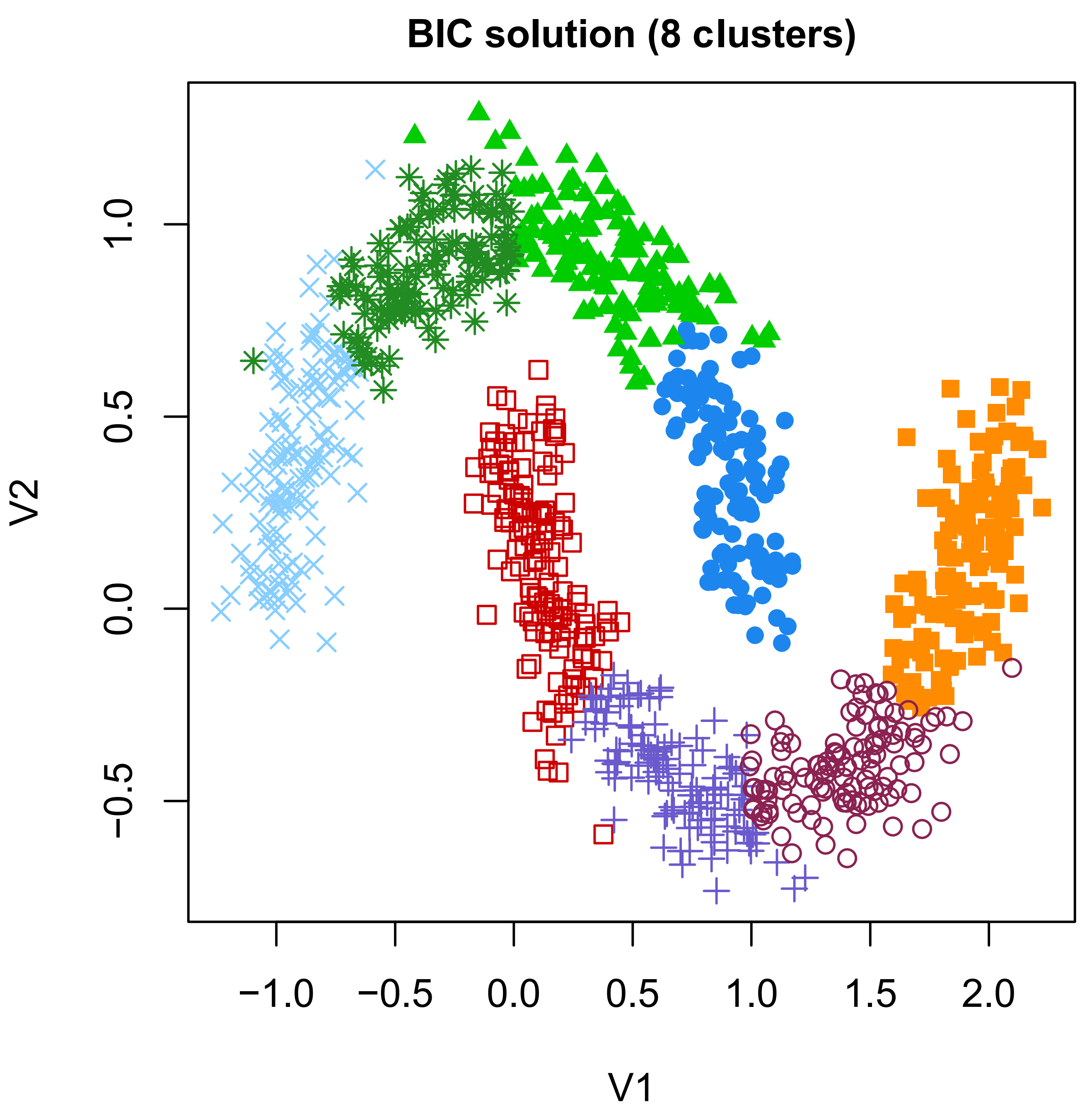
- Start out with the usual Gaussian mixture solution;
- **merge** “similar” *components* to create non-Gaussian *clusters.*

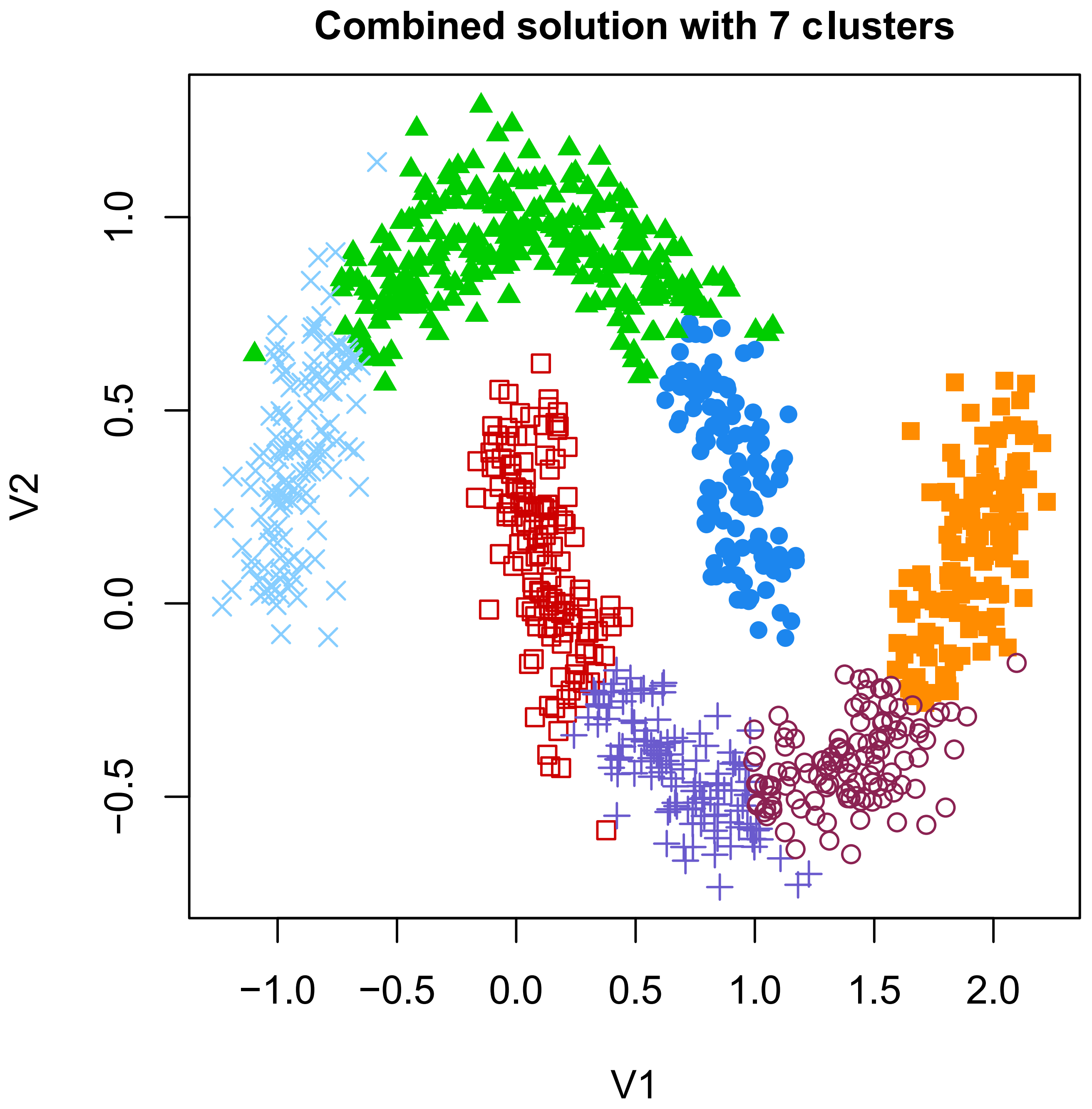
Note: we’re distinguishing “components” from “clusters” now.

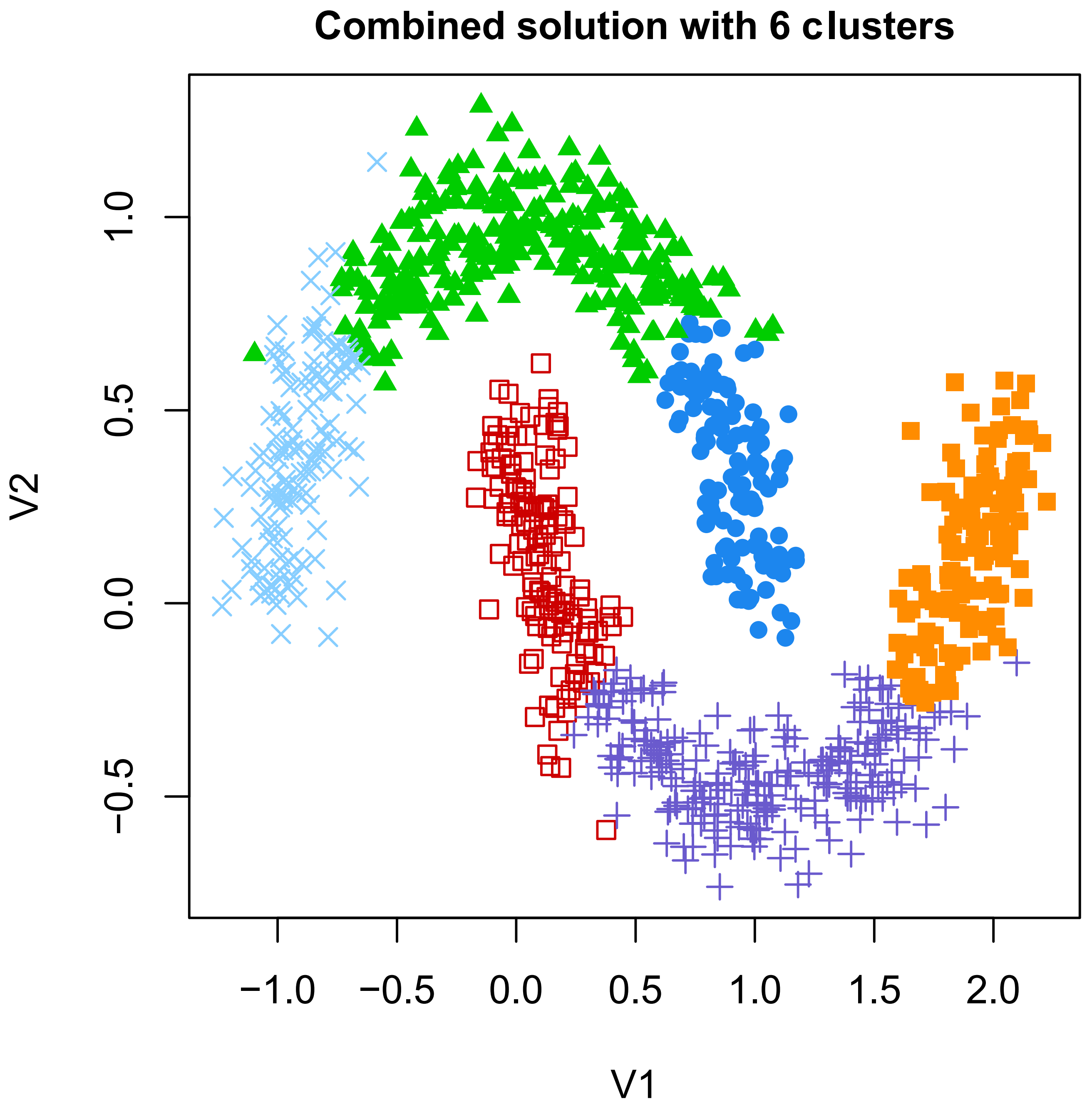
Merging components to get clusters

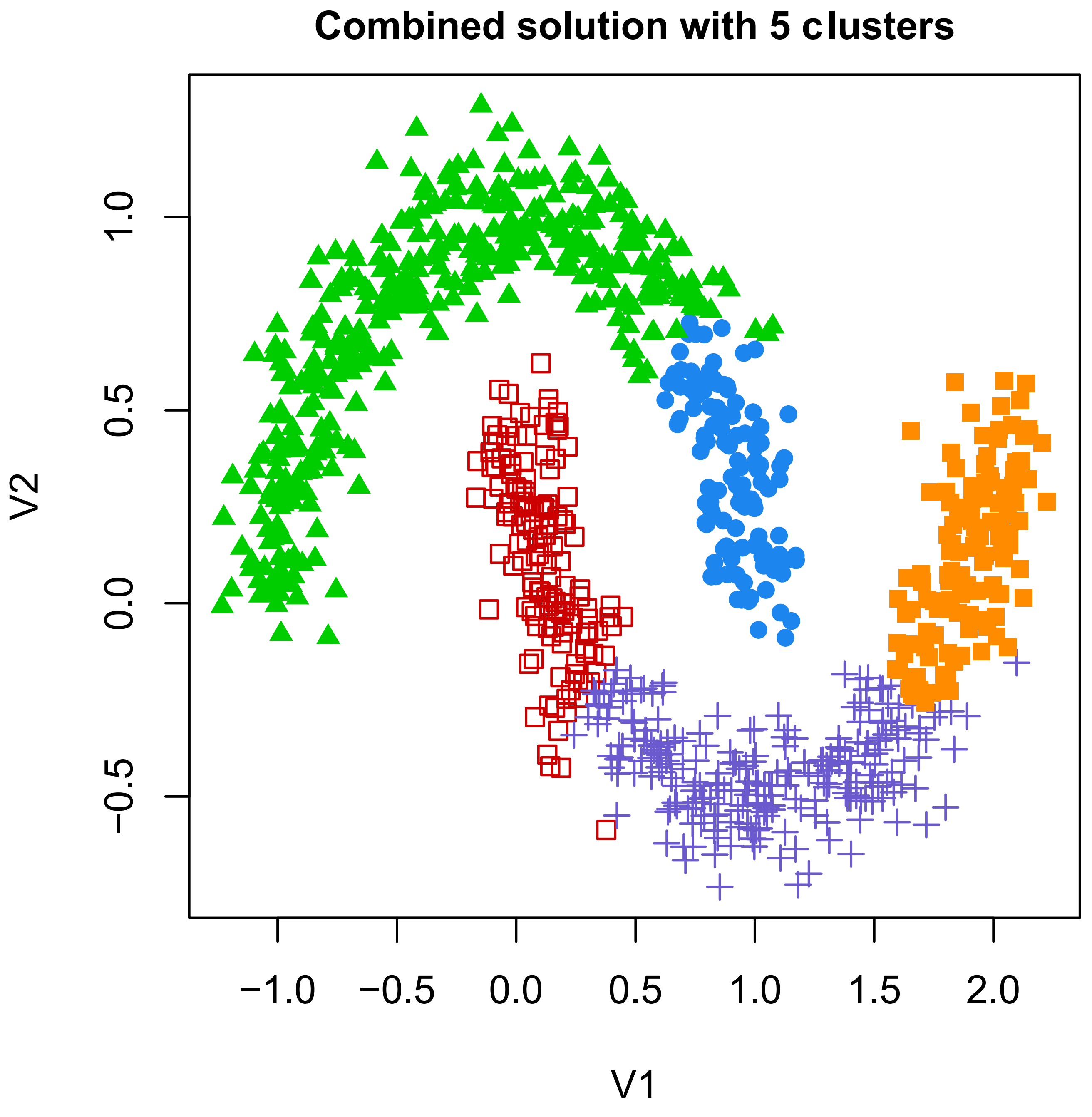
```
library(mclust)

output <- clustCombi(data = x)
plot(output)
```

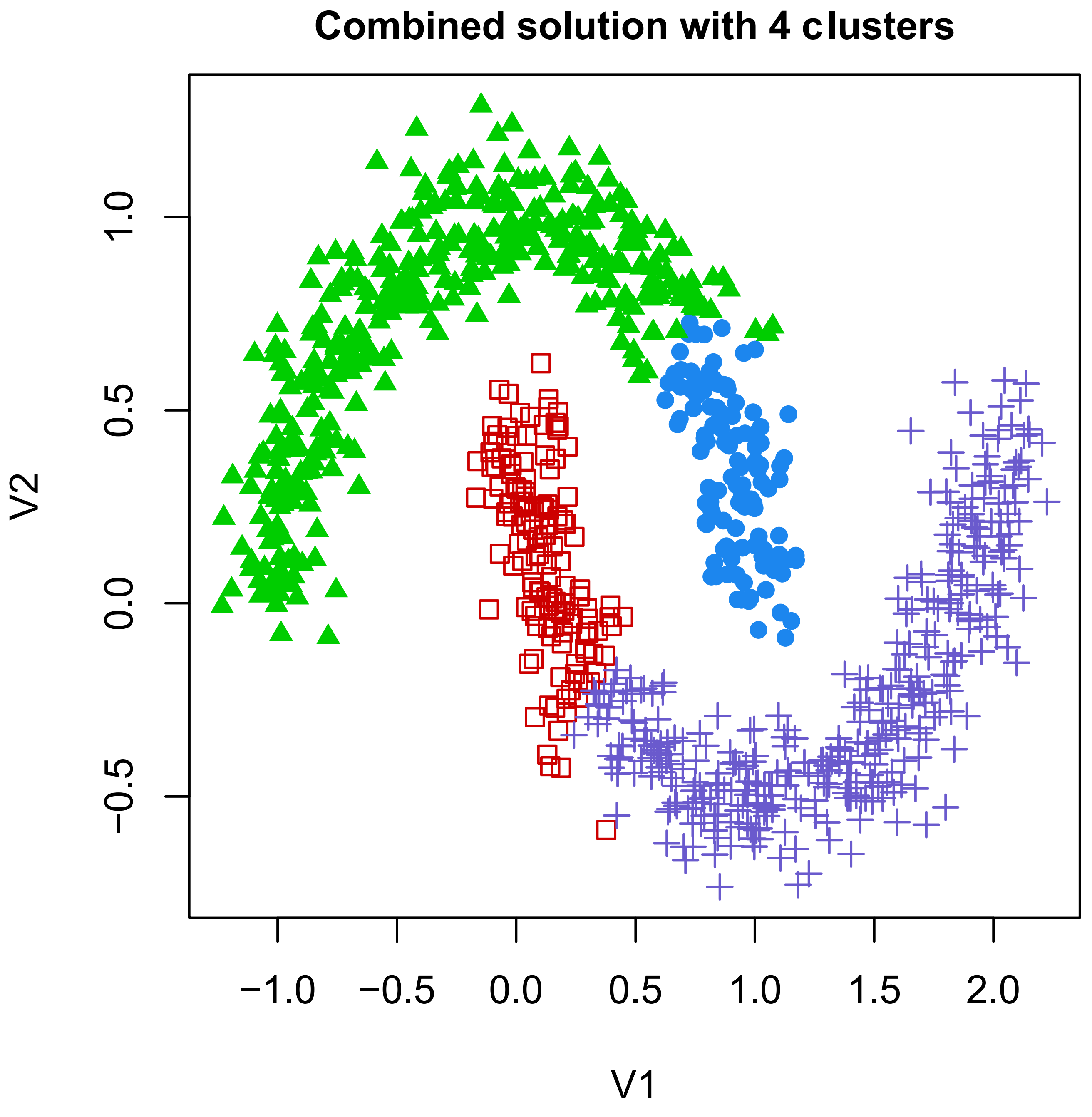


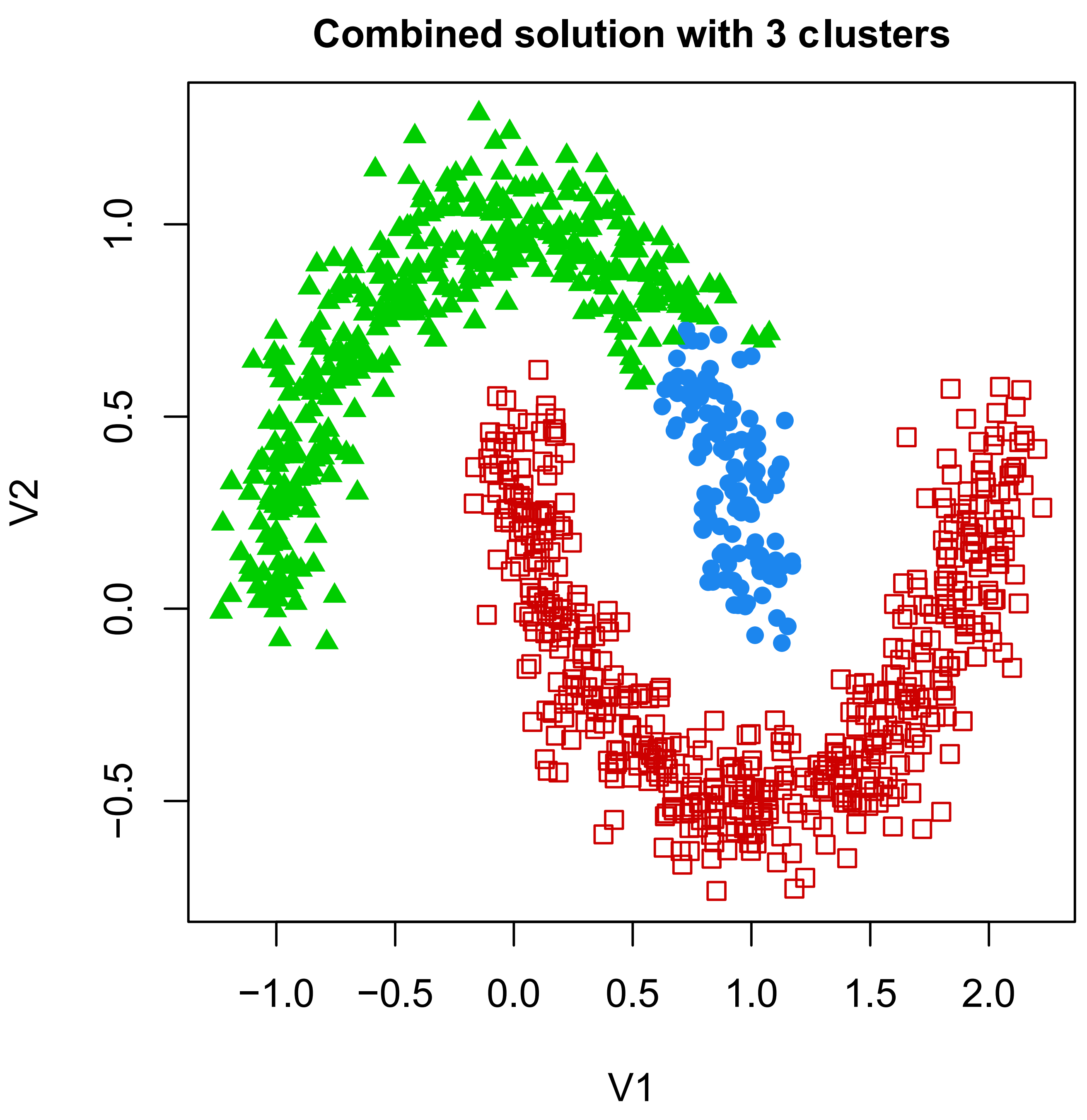

BIC solution (8 clusters)
V2
V1
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

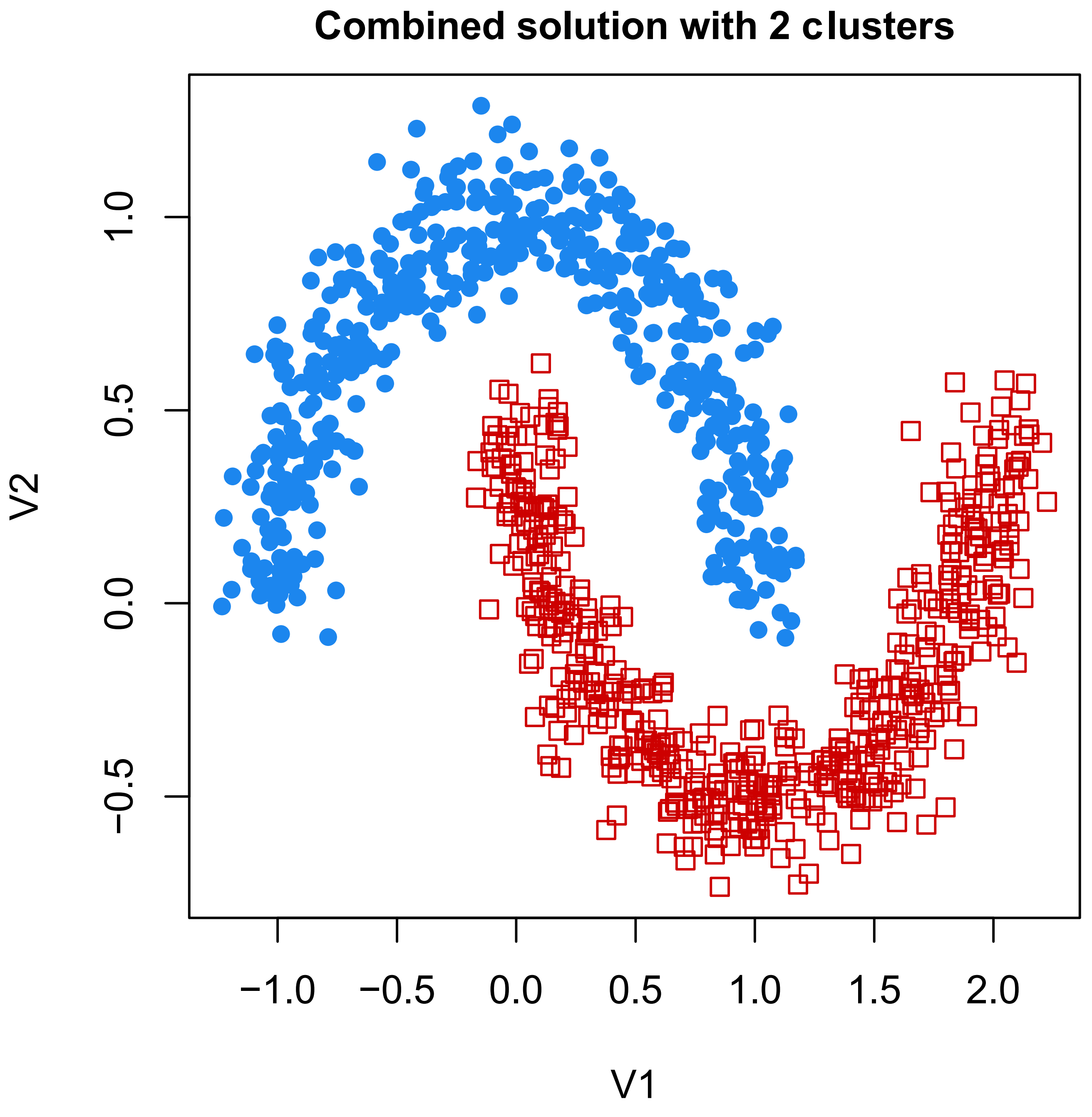

Combined solution with 7 clusters
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1

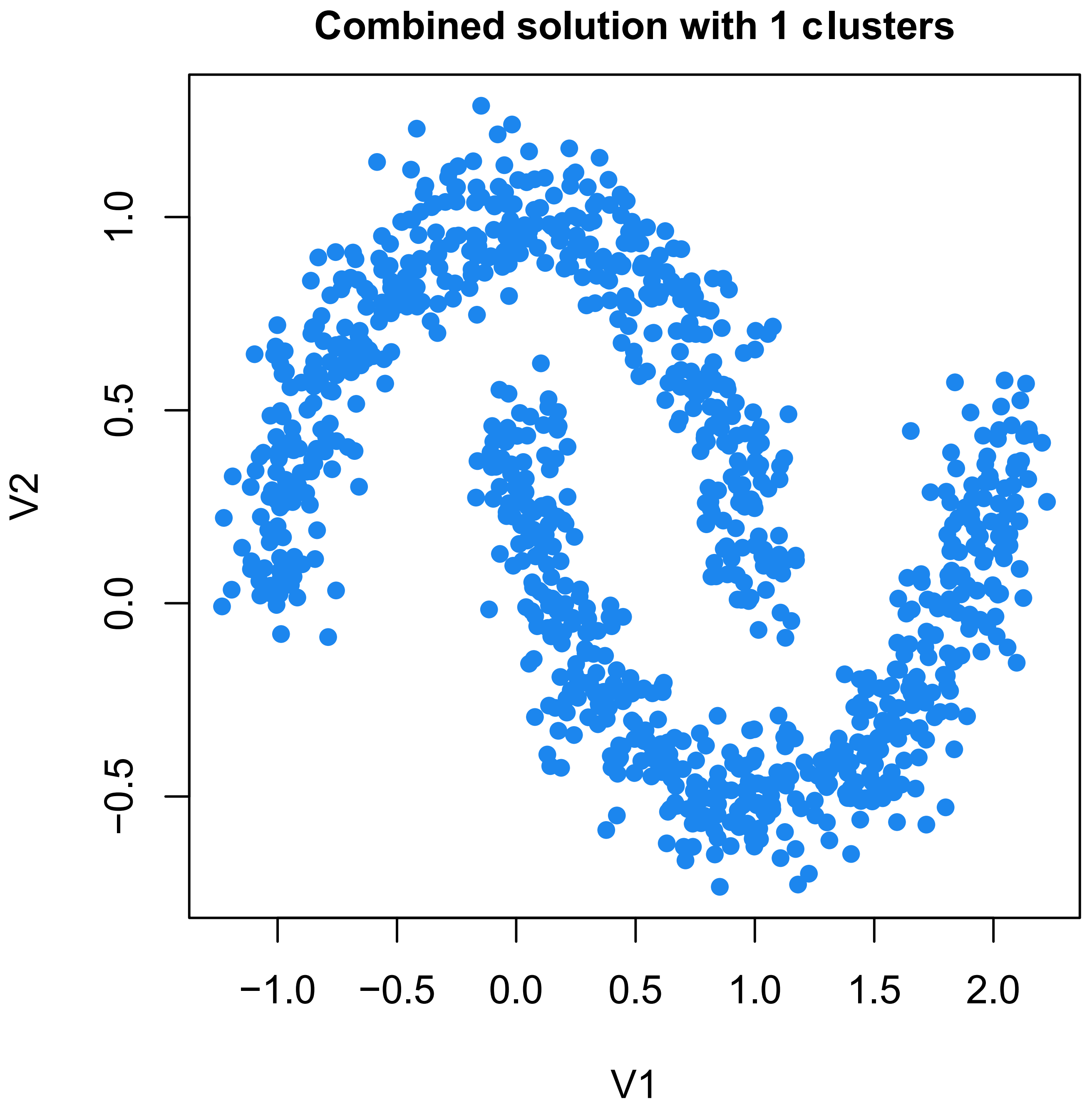

Combined solution with 6 clusters
V2
V1
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

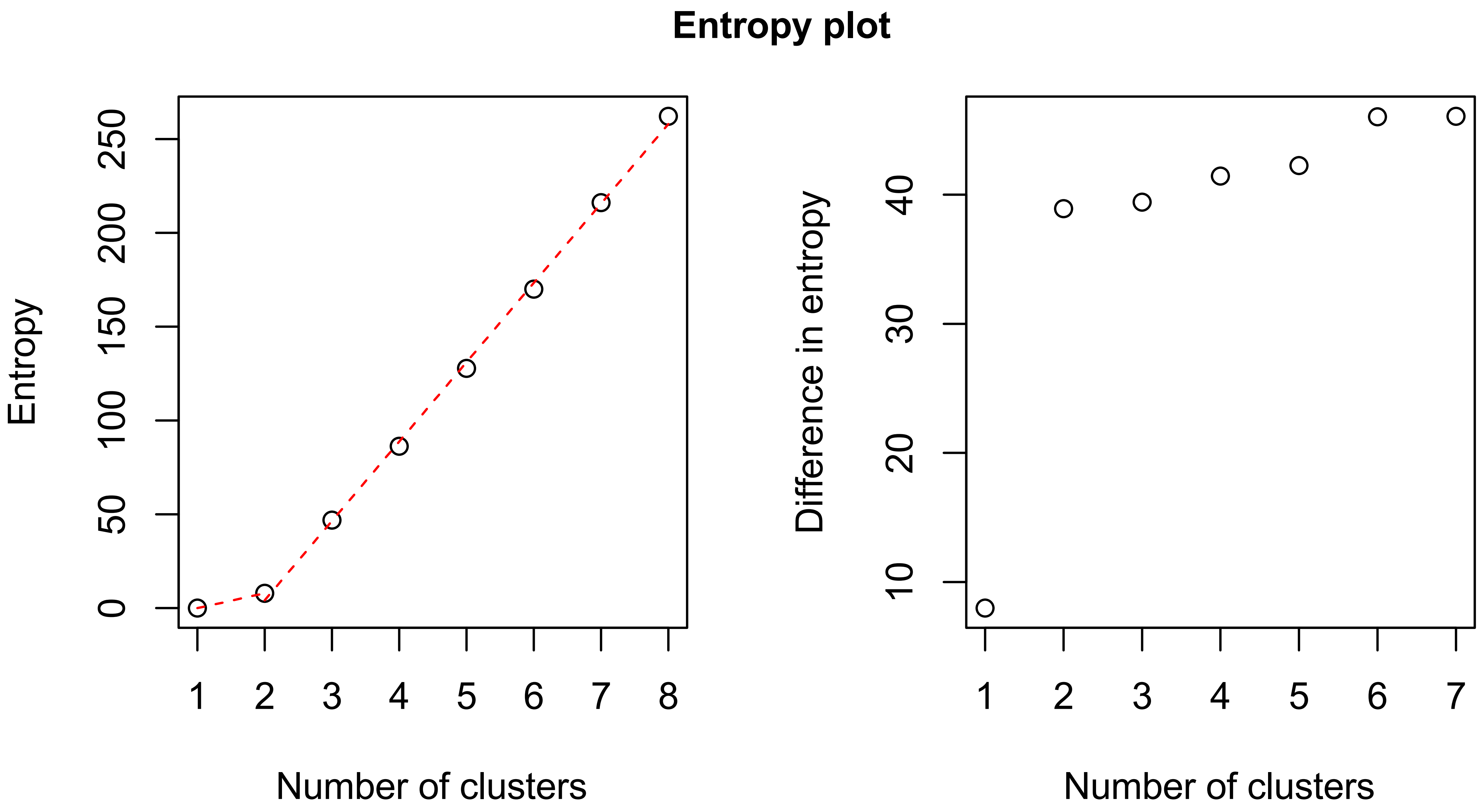

Combined solution with 5 clusters
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1


Combined solution with 4 clusters
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1


Combined solution with 3 clusters
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1

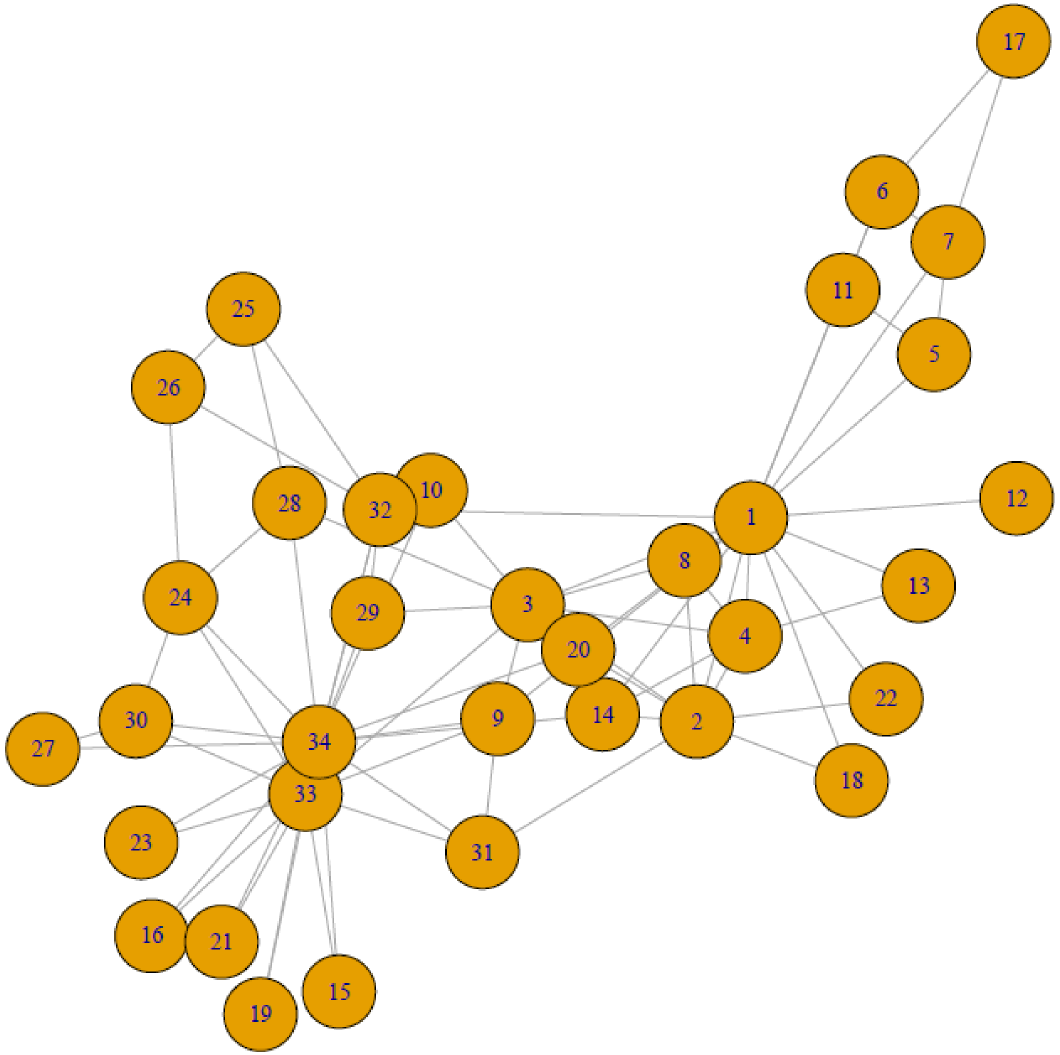

Combined solution with 2 clusters
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1


Combined solution with 1 clusters
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1


Entropy plot
Entropy
0
50
100
150
200
250
1
2
3
4
5
6
7
8
Number of clusters
Difference in entropy
10
20
30
40
1
2
3
4
5
6
7
Number of clusters

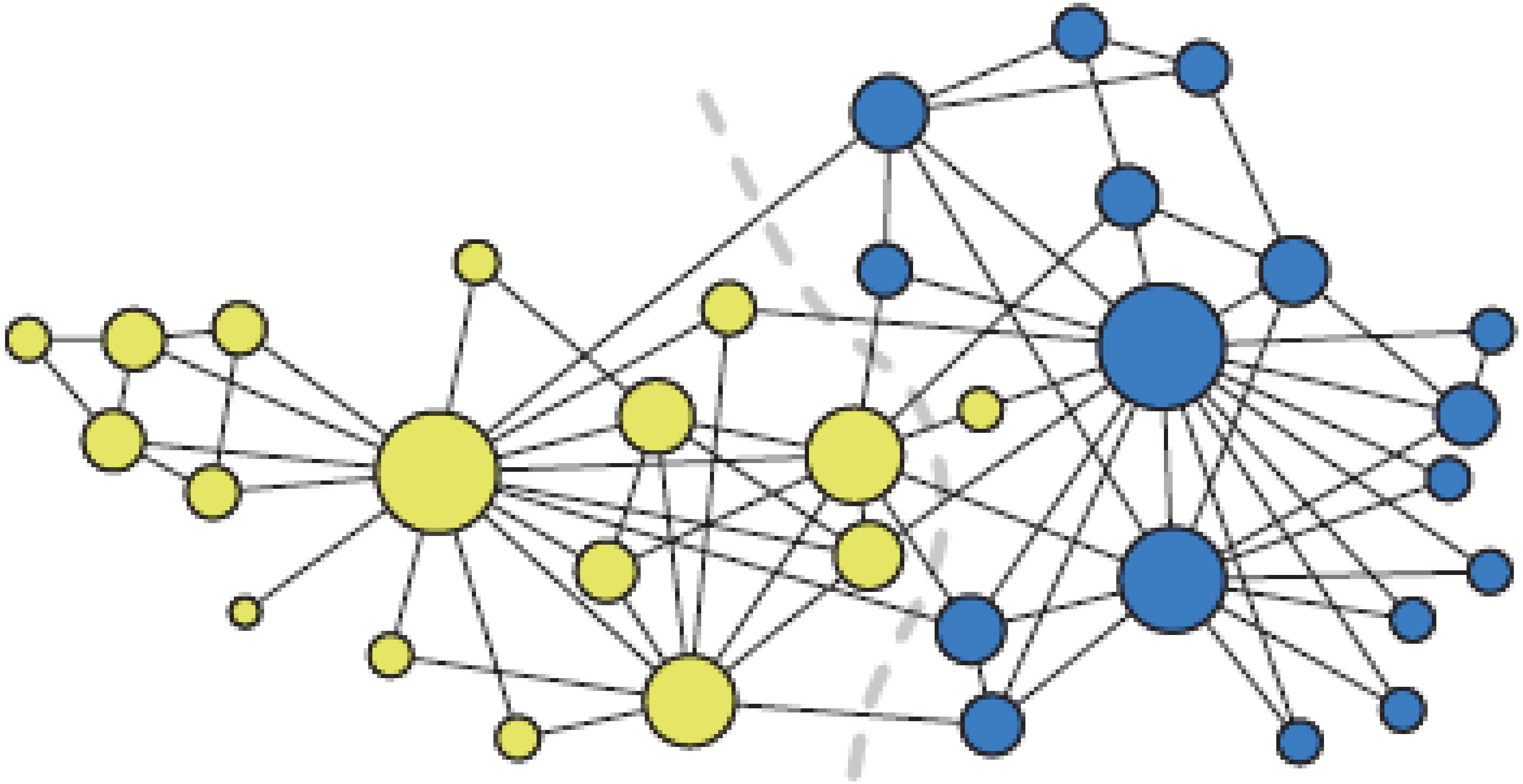
Clustering other types of things

Network data: Karate club



(Zachary, 1977)

Karate club “stochastic block model”



(Karrer & Newman 2010)

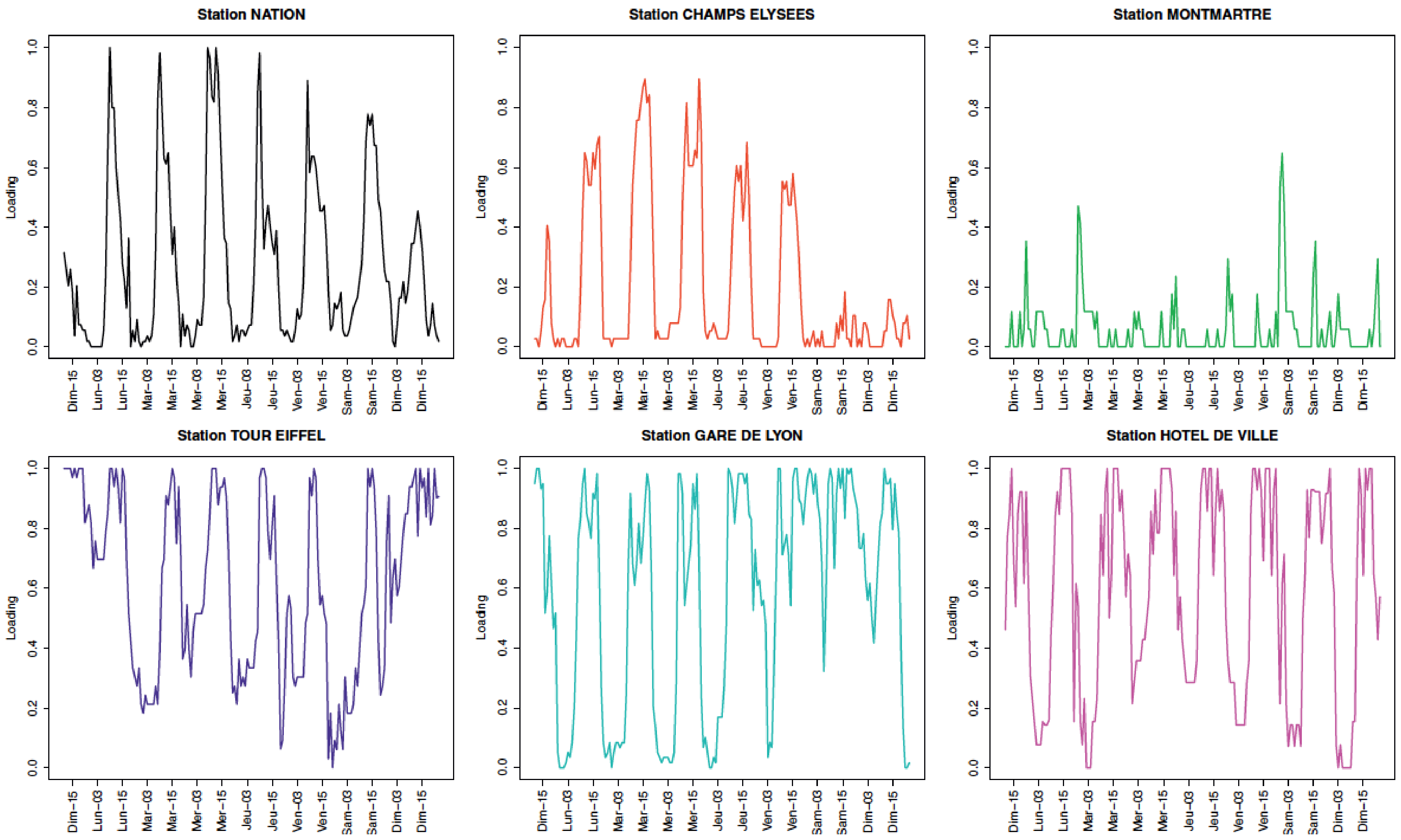
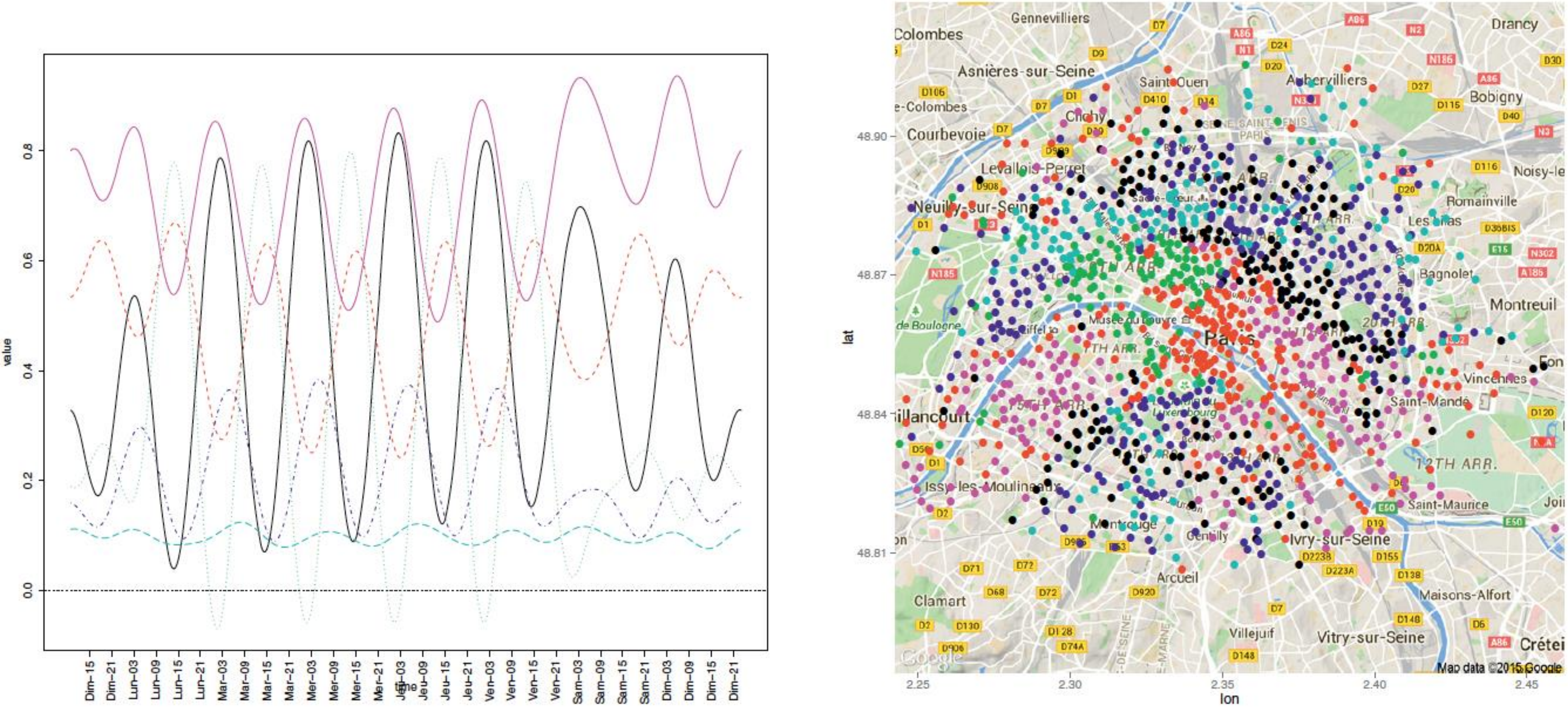


Figure 12.1 Loading profiles of some Vélib stations. A loading value equal to 1 means that the station is full of bikes whereas a value equal to 0 indicates a station without available bikes.

Bouveyron et al. 2018



(a) Cluster mean functions

(b) Map of clustered stations

Figure 12.6 Cluster mean functions and map of clustered stations by funFEM on the Vélib data set.

Conclusion

- **Model-based clustering:**
    1. Pretend we believe in a model;
    2. Estimate the model.
- Algorithm is defined by the model;
- Easy to think about assumptions;
- Flexible in using other data types;
- Common model: GMM (implementation `mclust` in `R`);
- Secret weapon: component merging.