Data Wrangling and Data Analysis
Clustering

Daniel Oberski & Erik-Jan van Kesteren

Department of Methodology & Statistics

Utrecht University

This week

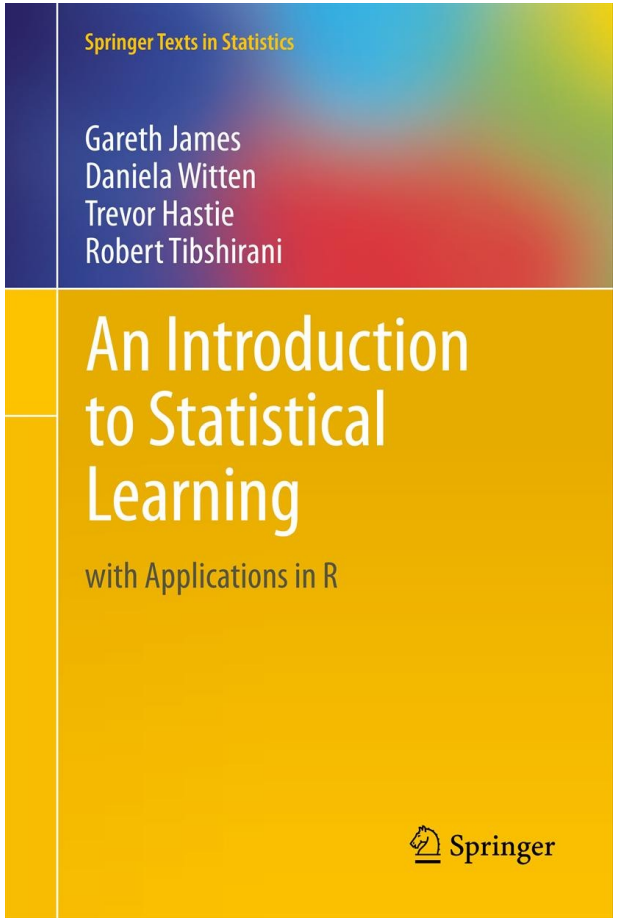
- Hierarchical vs. partitioned clustering
- The K-means algorithm
- **Model-based clustering**

Goal of the week: ***understand***, ***apply***, and ***evaluate*** clustering methods

Reading materials for this week

- Selected paragraphs from **Introduction to Statistical Learning (ISLR)** §10.3
- https://faculty.marshall.usc.edu/gareth-james/ISL/

- Mixture models: latent profile and latent class analysis (Oberski, 2016) §1.1, §1.2
- http://daob.nl/wp-content/papercite-data/pdf/oberski2016mixturemodels.pdf

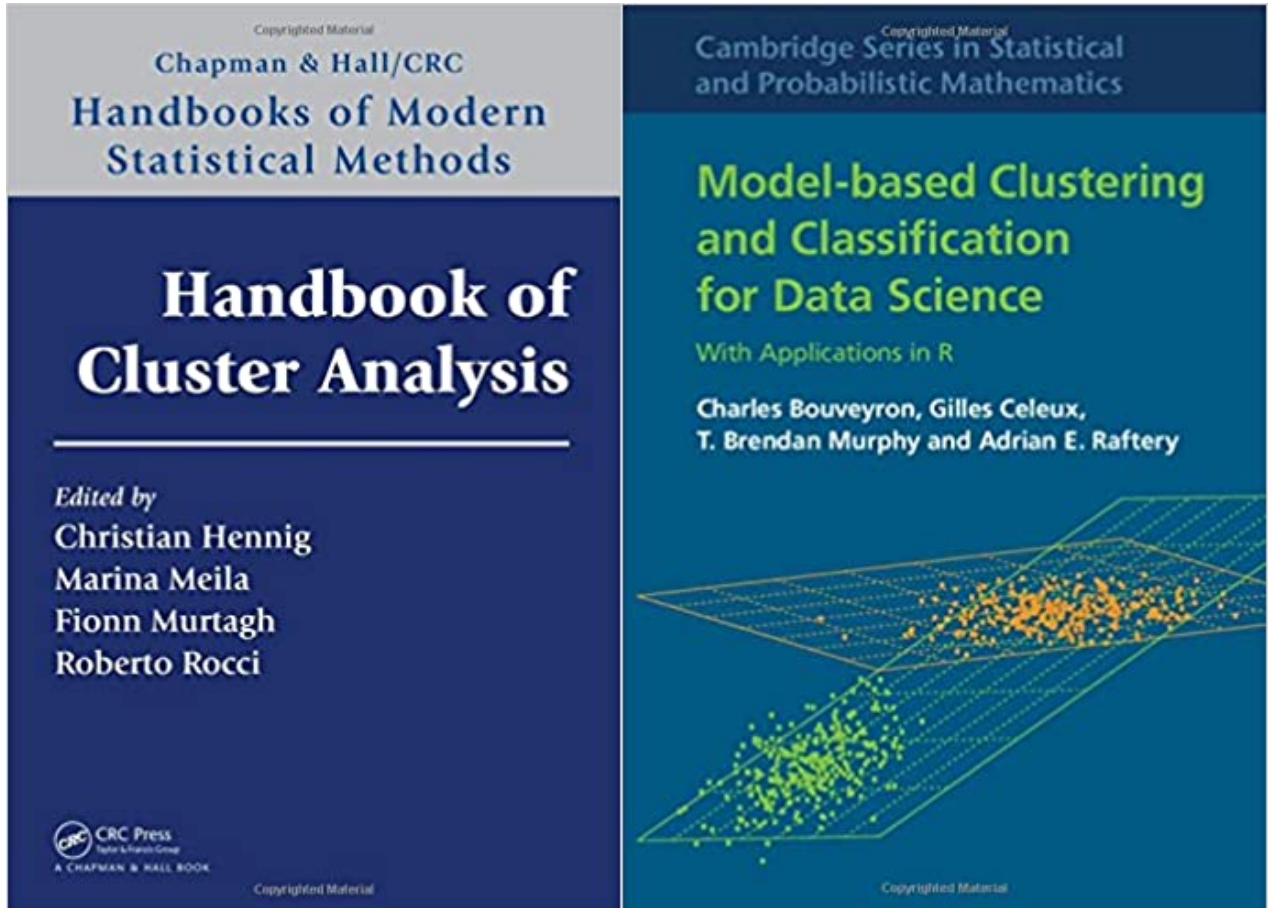


Optional, much more in-depth material

"Cluster validation: how to think and what to do"
– presentation by Hennig

Handbook of Cluster Analysis
Hennig et al. (2016)

Model-based Clustering and Classification for Data Science
Bouveyron et al. (2018)



Assignments this week

- Monday: R assignment on hierarchical and k-means clustering

- Tuesday: R assignment with mclust
- Thursday: either (a) resit for the test, or (b) programming k-medioids clustering from scratch

Model-based clustering

K-means, redux

- The default k-means algorithm makes K clusters which are: Circular in the space of the data. Is this reasonable?

- Maybe x and y covary within the clusters, in the same way or even differently? → ellipses

Model-based clustering

clustering especially interesting when it comes to exploring/finding new groups and not just to visualize groups
e.g. in medicine/patient trails finding new reactions groups

- Principled approach to clustering based on a statistical model
- Latent (unmeasured, unobserved) classes cause the observations
- Assumptions about the clusters are explicit, not implicit and ad-hoc:
  - The data within each cluster is normally distributed
  - The clusters may be equal or variable in the following components:
    - Volume (size of the clusters in data space)
    - Shape (circle or ellipse)
    - Orientation (the angle of the ellipse)

fuzzy clustering

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

bernoulli distribution if variable is 0/1

Model-based clustering

Other names for model-based clustering:

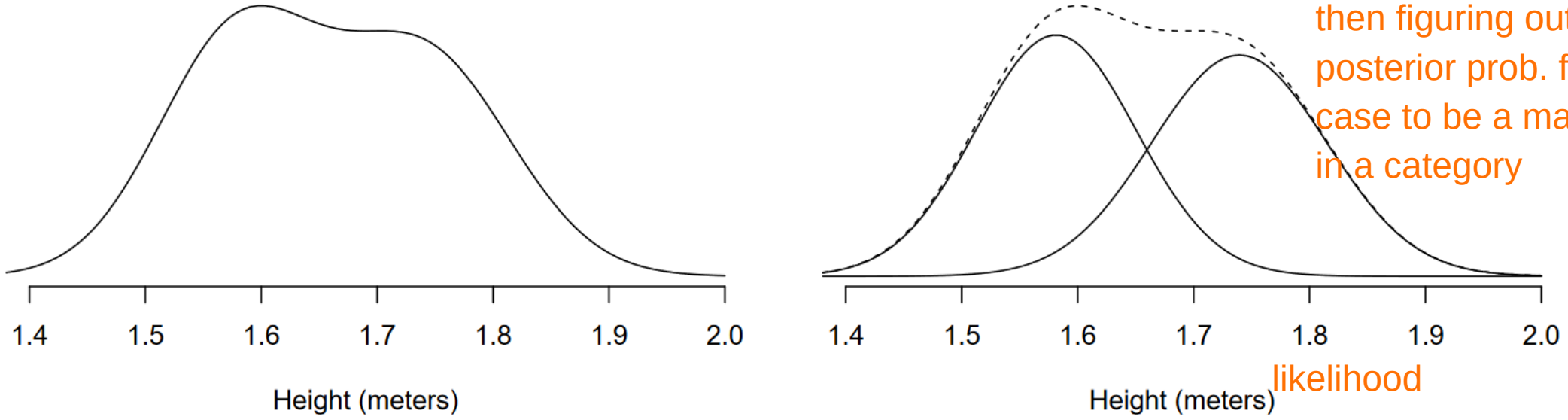
- (Gaussian) mixture models
- Latent profile analysis
- Latent class analysis (categorical observations)
- Latent Dirichlet allocation

Model-based clustering

Clustering is getting us from the big distribution including all data to two clusters where we can identify men and women within the data

two bumps because it's two groups

then figuring out the posterior prob. for each case to be a man/woman - in a category



likelihood

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

still - if one can define a distance - one can cluster it

Model-based clustering

probability of seeing the data point

- Statistical model + assumptions defines a likelihood

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

because we are trying to find the group proportions (mean, variance,...) to find the maximum likelihood for each data point

- Maximum likelihood estimation: find the parameters θ for which it is most likely to observe this data y
- This is how models can be estimated / fit / trained

- NB: the model and its assumptions are debatable!

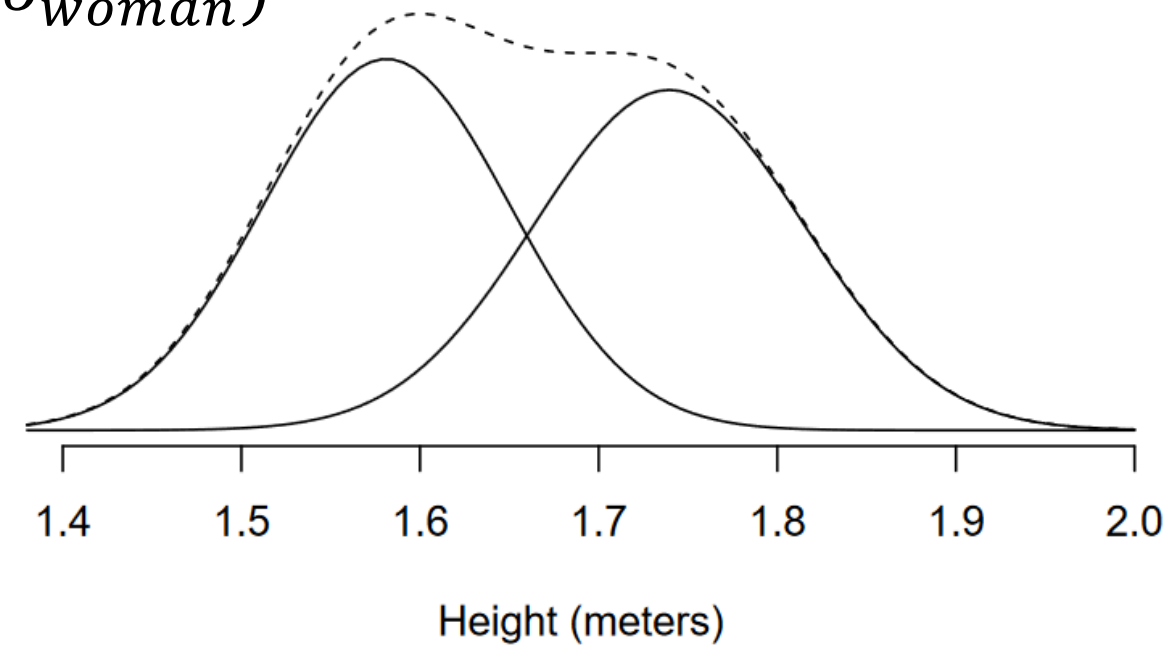
Model-based clustering

Likelihood (*density*) for height data:

$$p(\text{height} \mid \theta) =$$
$$Pr(\text{man})\text{Normal}(\mu_{man}, \sigma_{man}) +$$
$$Pr(\text{woman})\text{Normal}(\mu_{woman}, \sigma_{woman})$$

Or, in clearer notation:

$$p(\text{height} \mid \theta) =$$
$$\pi_1^X \text{Normal}(\mu_1, \sigma_1) +$$
$$(1 - \pi_1^X)\text{Normal}(\mu_2, \sigma_2)$$



min 40 day 2

Model-based clustering

difference to k-Clustering
- the spread of the data can be different and doesn't have to be the same

Gaussian mixture parameters:

- π_1^X determines the relative cluster sizes
  - how many observations / examples are to be expected in each cluster)
- μ_1 and μ_2 determine the locations of the clusters
  - These are equivalent to the 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 parameters describe our model of how the data is generated!

Estimation: the EM algorithm

if we knew it exactly

- If we know who is a man and who is a woman, it's easy to find the maximum likelihood estimates for μ and σ:

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

creating a weighted mean and not just the mean

- But we don't know this! Assignments need to be estimated too!

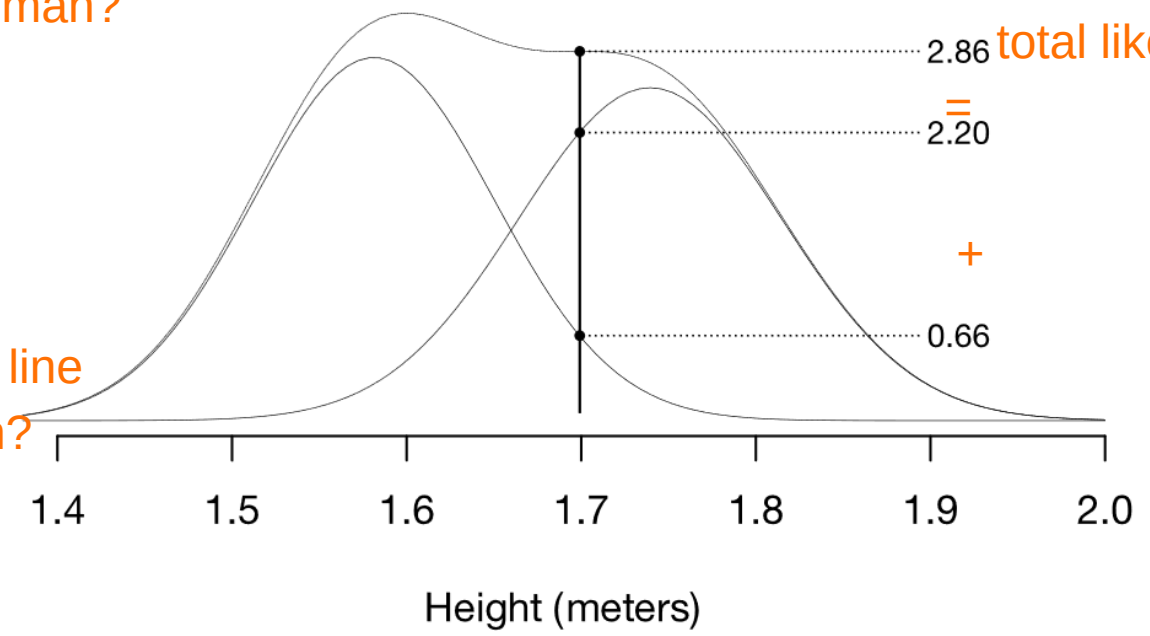
Estimation: the EM algorithm

- 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?

what's the prob. for the parameters for it to be a man/woman?

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

what proportion of the line is made up out of men?



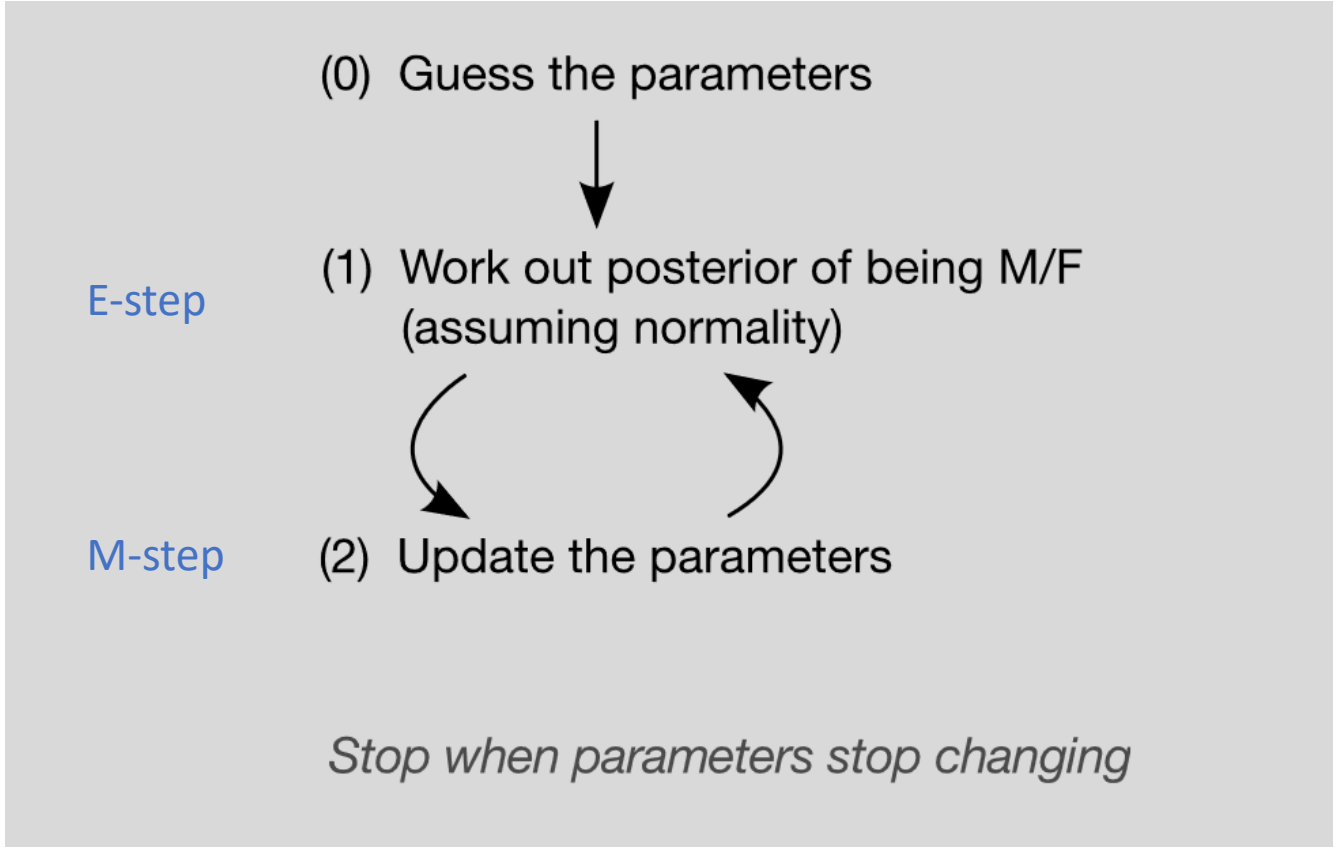
Estimation: the EM algorithm

- Then we know the class assignments (probabilities)!
- Now can go back to the parameters and update them using maximum likelihood (M-step)
- Then, we can compute new posterior probabilities (E-step)

- NB: very, very similar to k-means but more formally defined

parameters are modified which changes the curve and then it keeps going back and forth until the change is very small or the model doesn't improve anymore

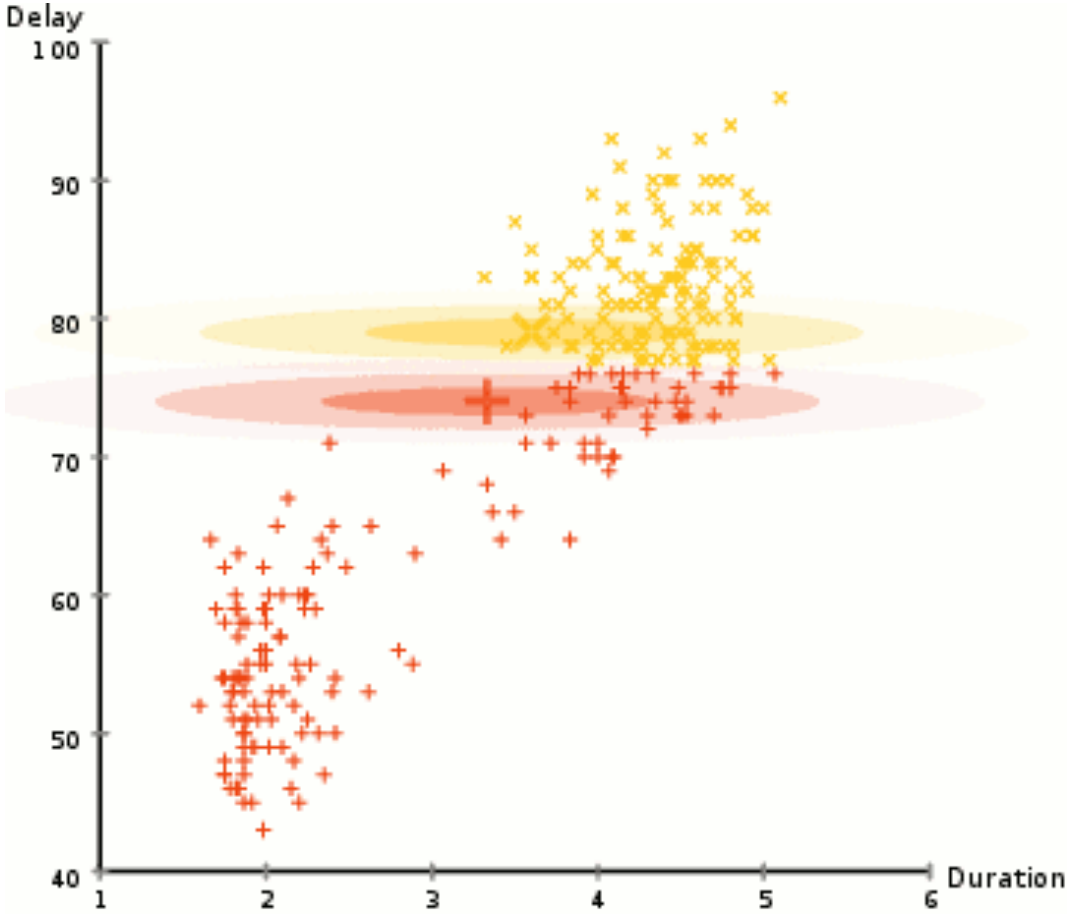
Estimation: the EM algorithm



e.g. prob. of the height of a woman is 1.6

and then she's 1.7m - so I update the posterior

Estimation: the EM algorithm

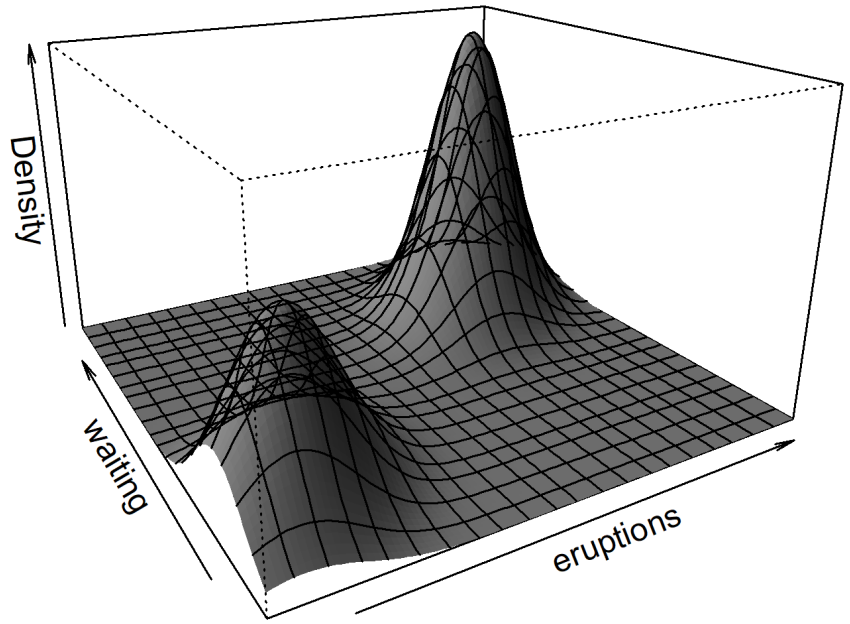


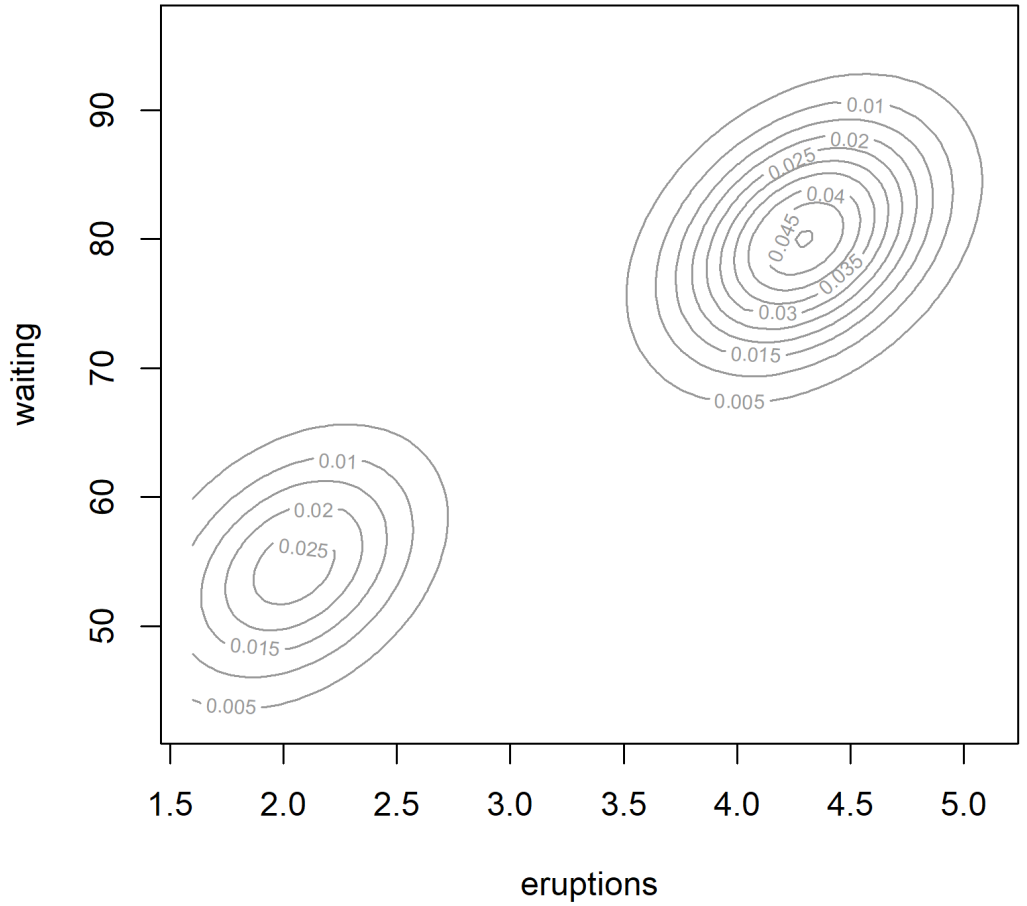
Multivariate model-based clustering

- With 2 observed features: that's why its an ellipse and not a curve anymore
  - 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

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





numer of classes is an assumption like in k-means

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)

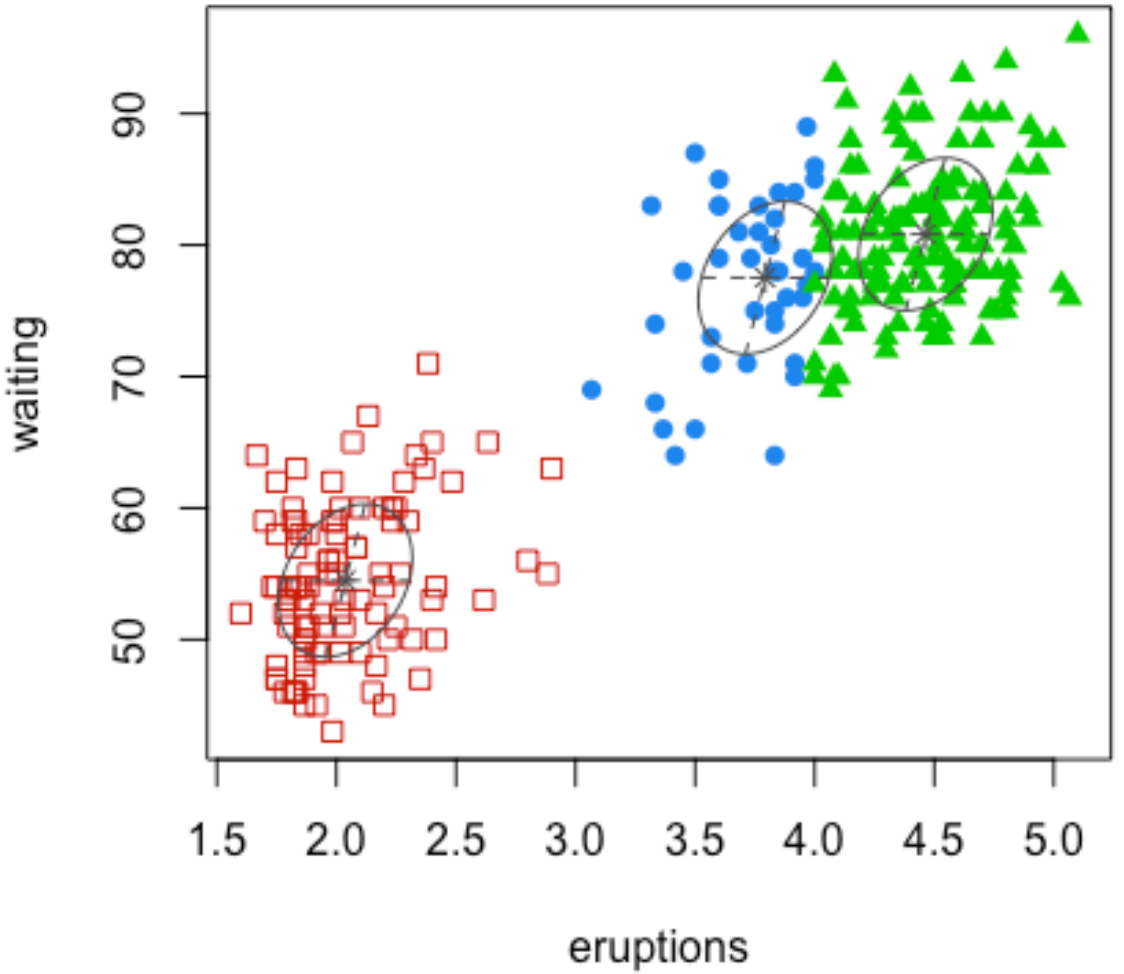
Number of parameters

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

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
  - =Exactly probabilistic *K*-means!
- Can also be different across clusters
  - not possible in k-means
- More flexible, complex model
  - Think about the 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

File size increases with number clusters

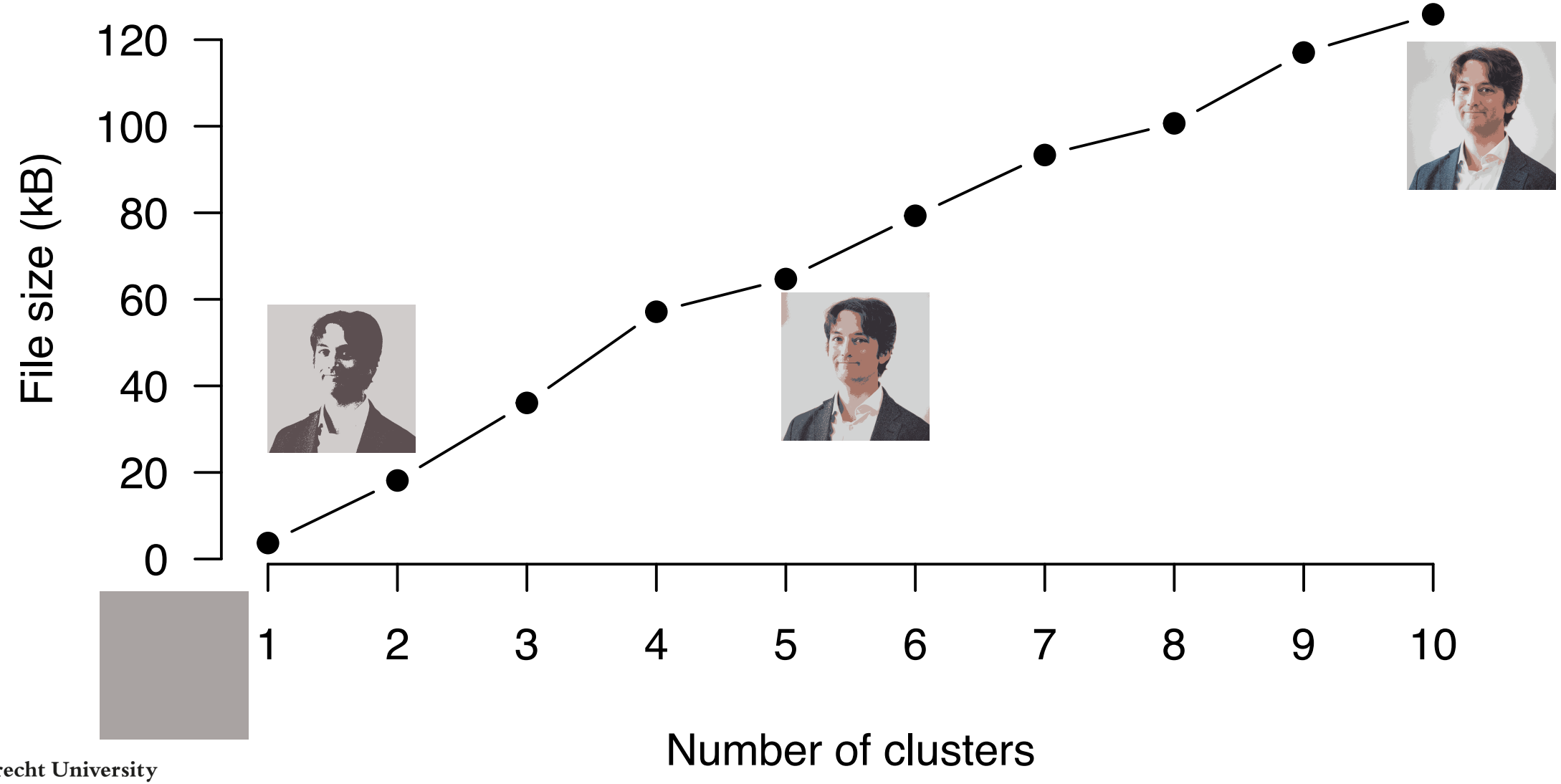
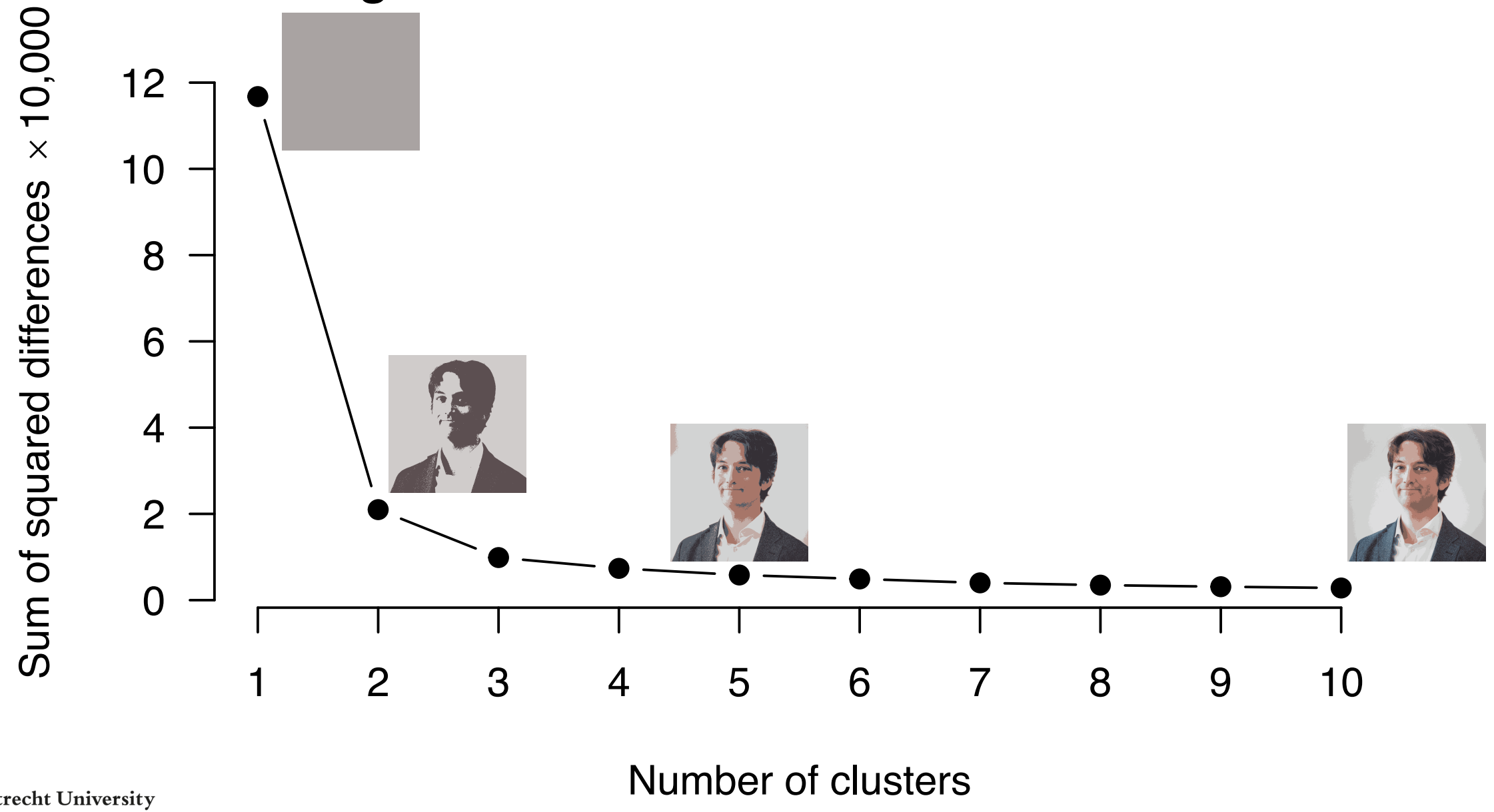
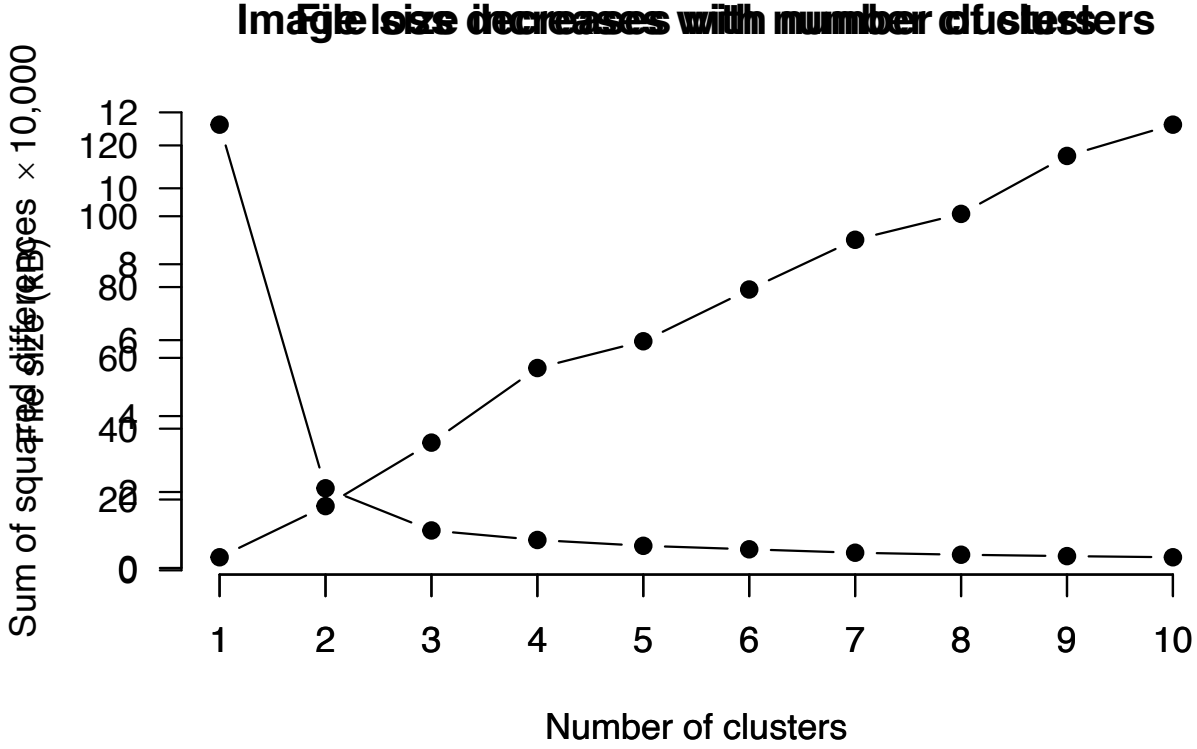


Image loss decreases with number of clusters





- More clusters gives **better "fit"** in terms of reconstruction of the image
  (compression is less "lossy")
- More clusters gives **bigger file size**
  (solution is more complex, takes more bytes to store)
- So the **model loss and model complexity trade off against each other**
- This is a common theme in (unsupervised) machine learning and you should remember this for model-based clustering lecture

Model fit

how well does the model perform

- 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, $p(\text{data} \mid \theta)$ bigger likelihood is better
- $-2 \cdot \log(\ell)$: *"Deviance"* negative because turned into a "loss" - the more classes the better the loss but also the more parameters (complexity goes up)

loss based on the likelihood = deviance

- m : Number of parameters
- n : Number of observations/examples

Model fit

- Tradeoff between fit and complexity

$$\underbrace{-2 \cdot \log(\ell)}_{\text{“Reconstruction loss”}} + \underbrace{m \cdot \log(n)}_{\approx\text{“File size”}^*}$$

approximation of file size

- Think: bias and variance tradeoff
  - Variance also has to do with “clustering stability”

- Better fit *and* lower complexity = better cluster solution

*Approximation for BIC, different choices possible

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

- `mclust` implements multivariate model-based clustering
- Provides an easy interface to fit several parameterizations
- Model comparison with BIC
- Plotting functionality

gaussian especially

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

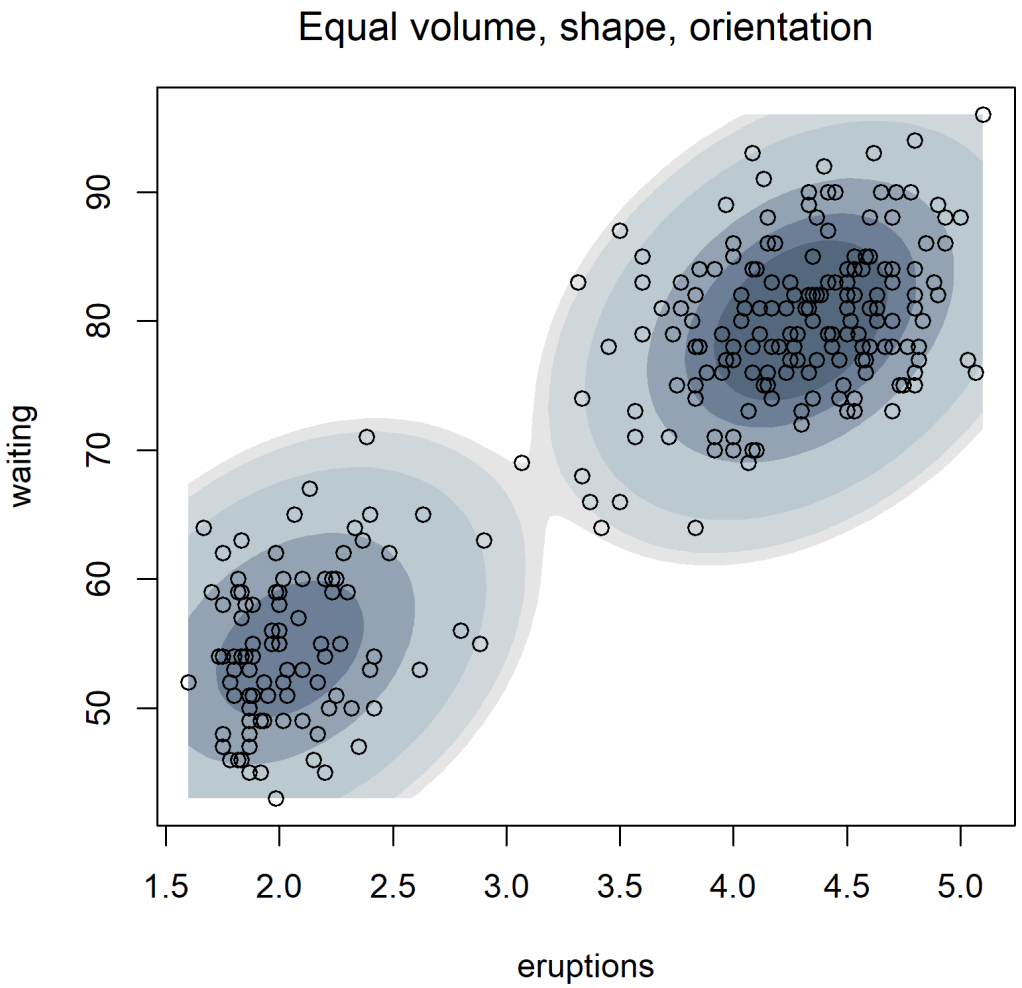
Model-based clustering in R

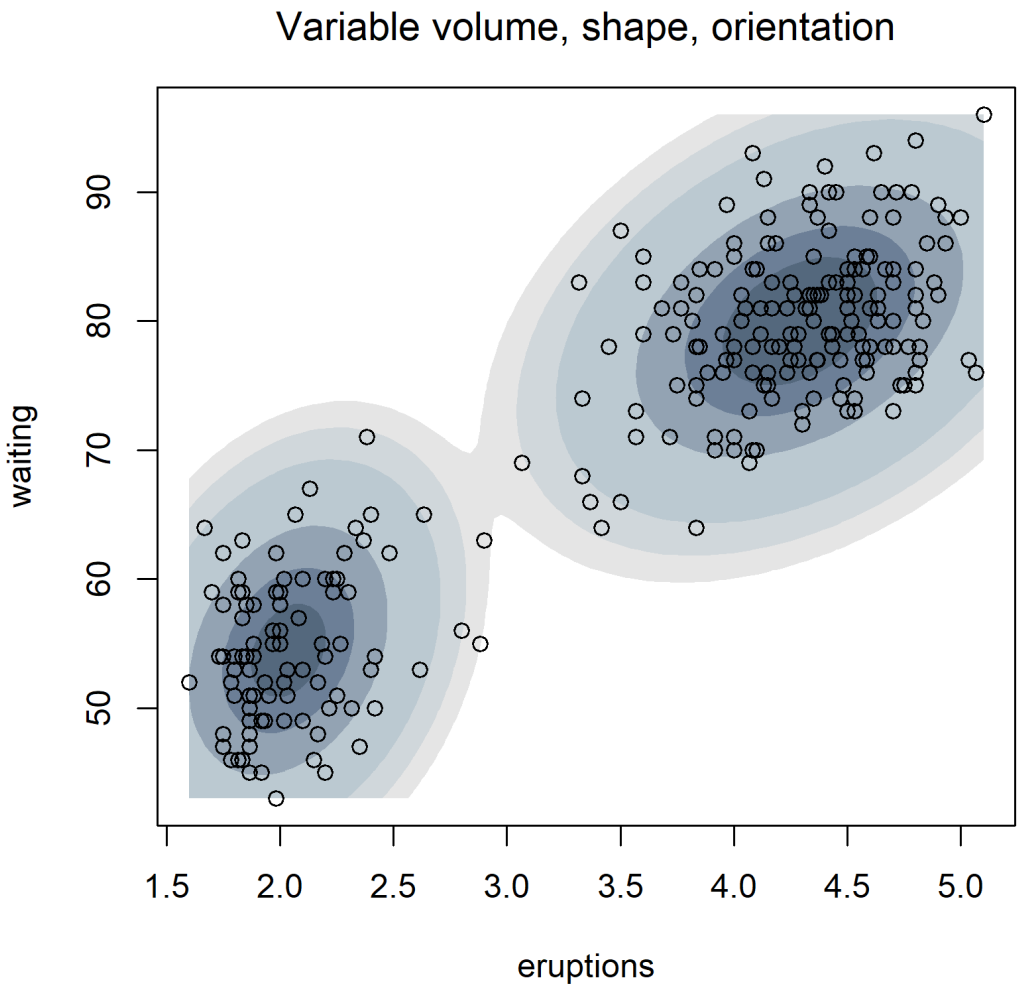
- Mclust uses an identifier for each possible parametrization :
- **E** for **e**qual, **V** for **v**ariable, **I** for identity matrix:

  - 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

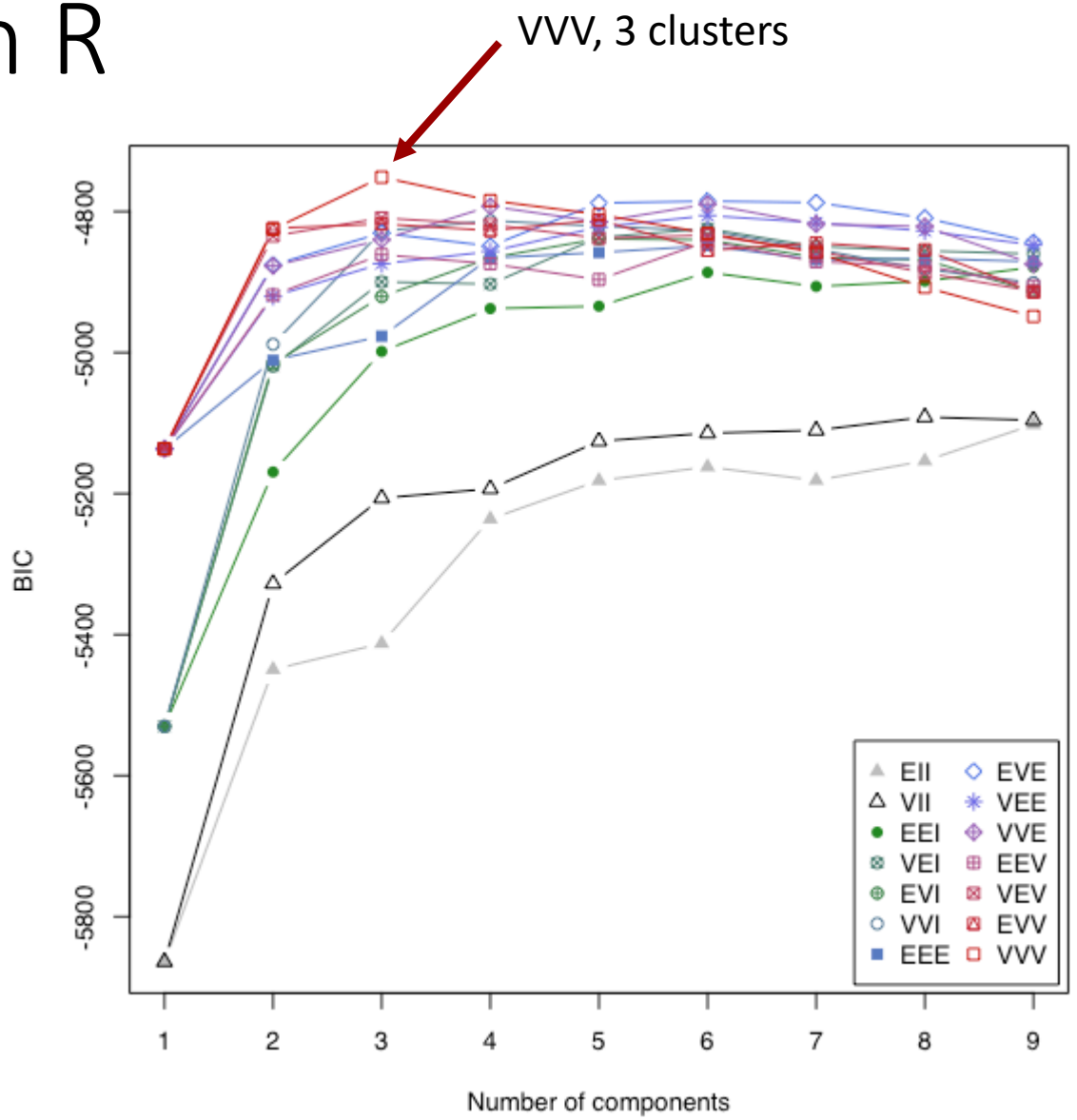




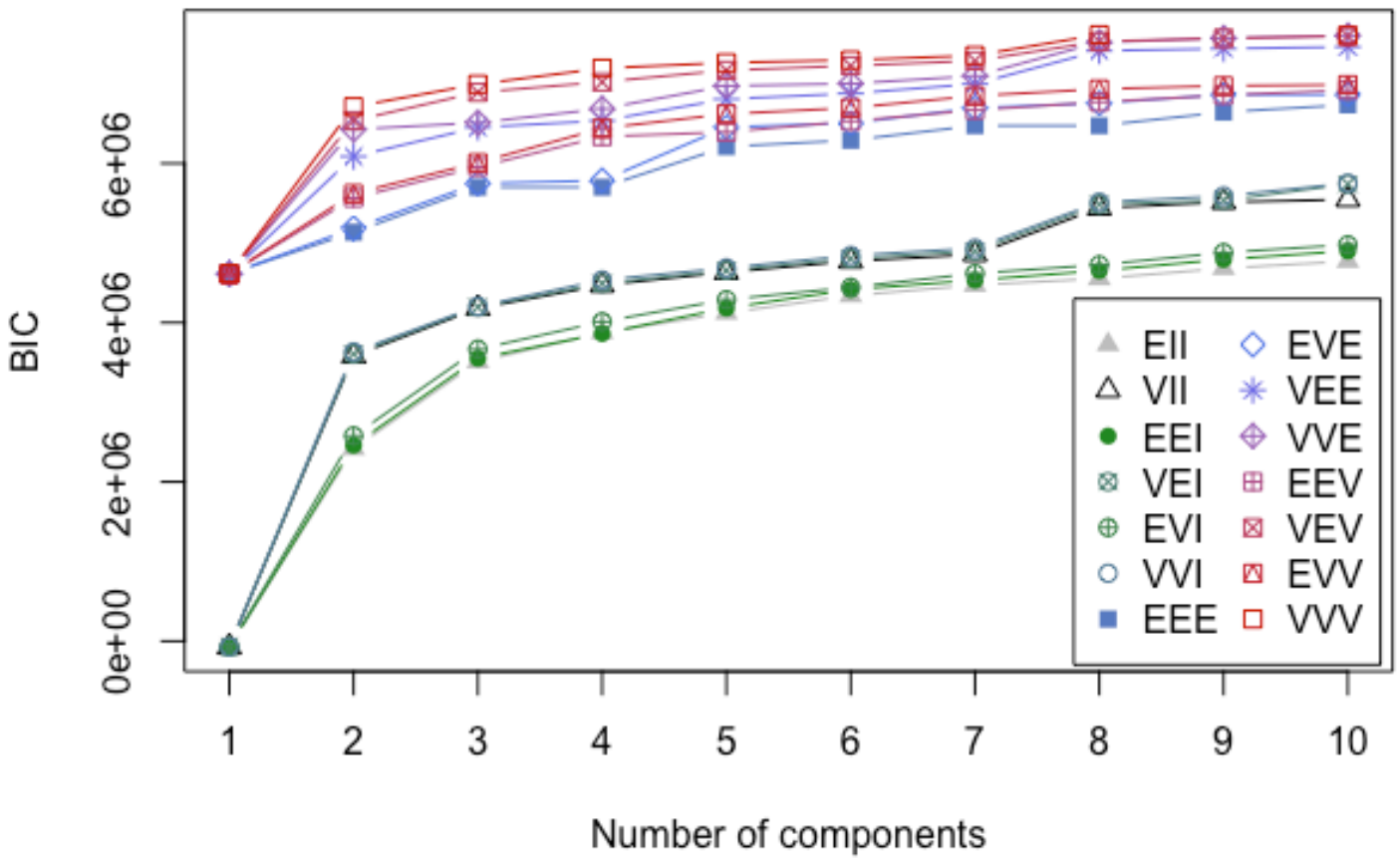
Model-based clustering in R

- 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



Model selection using BIC for image example



took 45 minutes

```
> fit_mc <- Mclust(im_ar, G = 1:10)
fitting ...
  |=============================================================| 100%
> summary(fit_mc)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------
Mclust VVV (ellipsoidal, varying volume, shape, and orientation)
model with 8 components:

 log-likelihood      n df     BIC     ICL
        3808542 640000 79 7616028 7530927

Clustering table:
     1      2      3      4      5      6      7      8
151032  48661 155542  34602  82621  49494  41665  76383
```

Merging simple components to get clusters

- Gaussian mixture modeling says each component should be Gaussian
- What if data are not Gaussian?

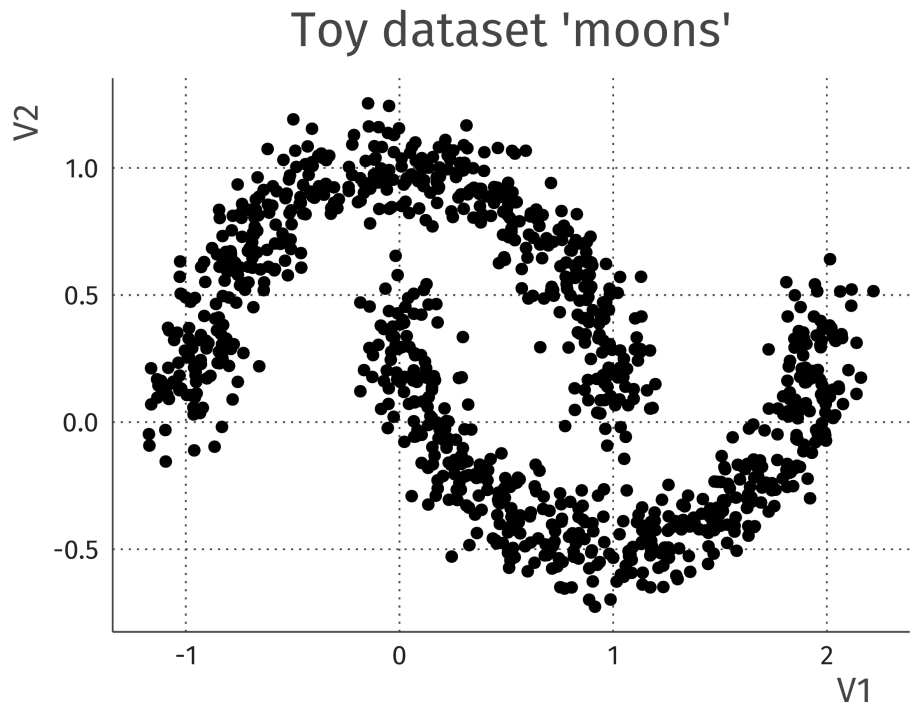
- **Useful idea**:
  - Start out with the usual Gaussian mixture solution
  - **merge** *components* that are “similar” to create the non-Gaussian *clusters*

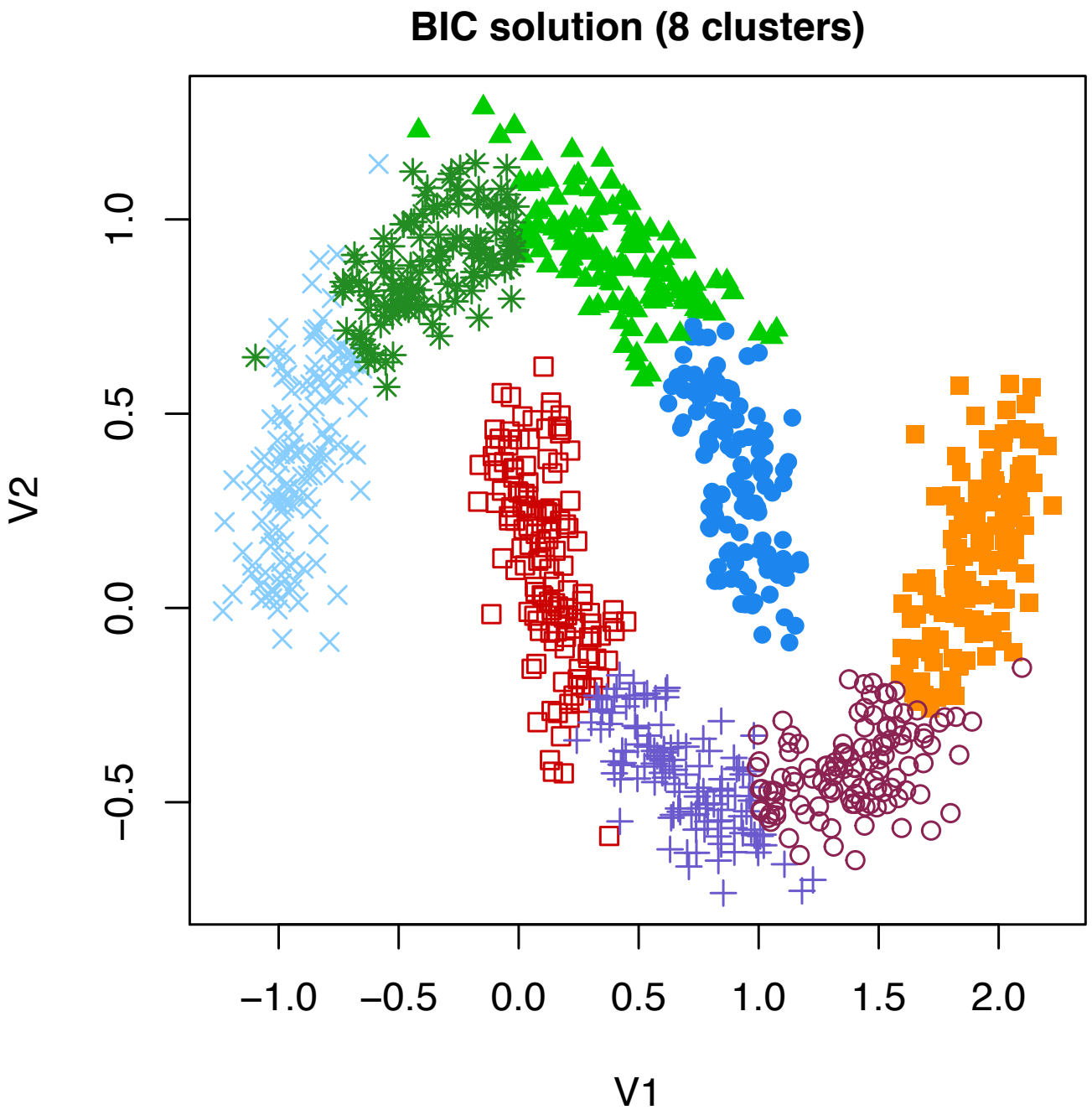
- *Note we’re distinguishing “components” from “clusters” now!*

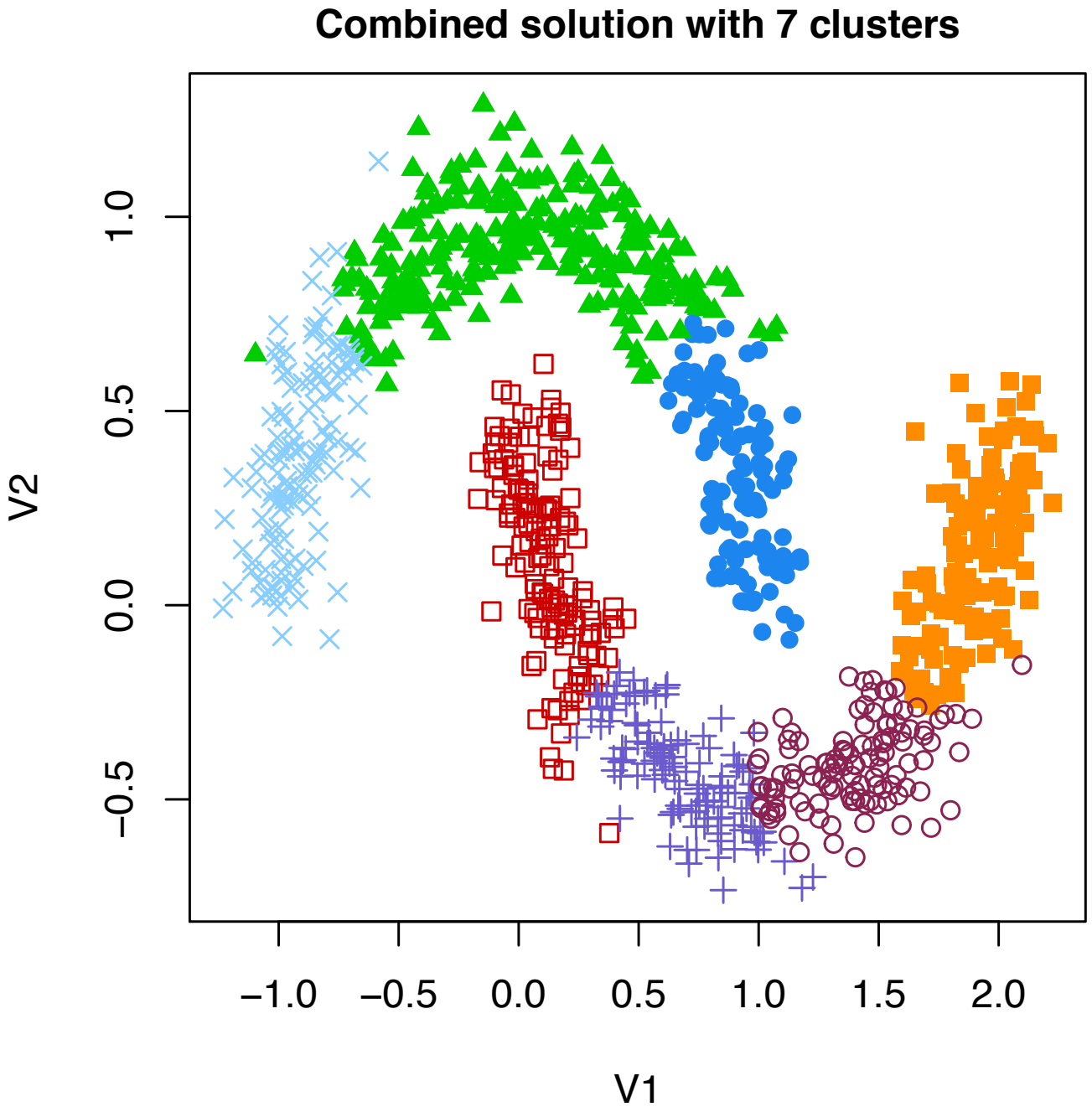
Merging simple components to get clusters

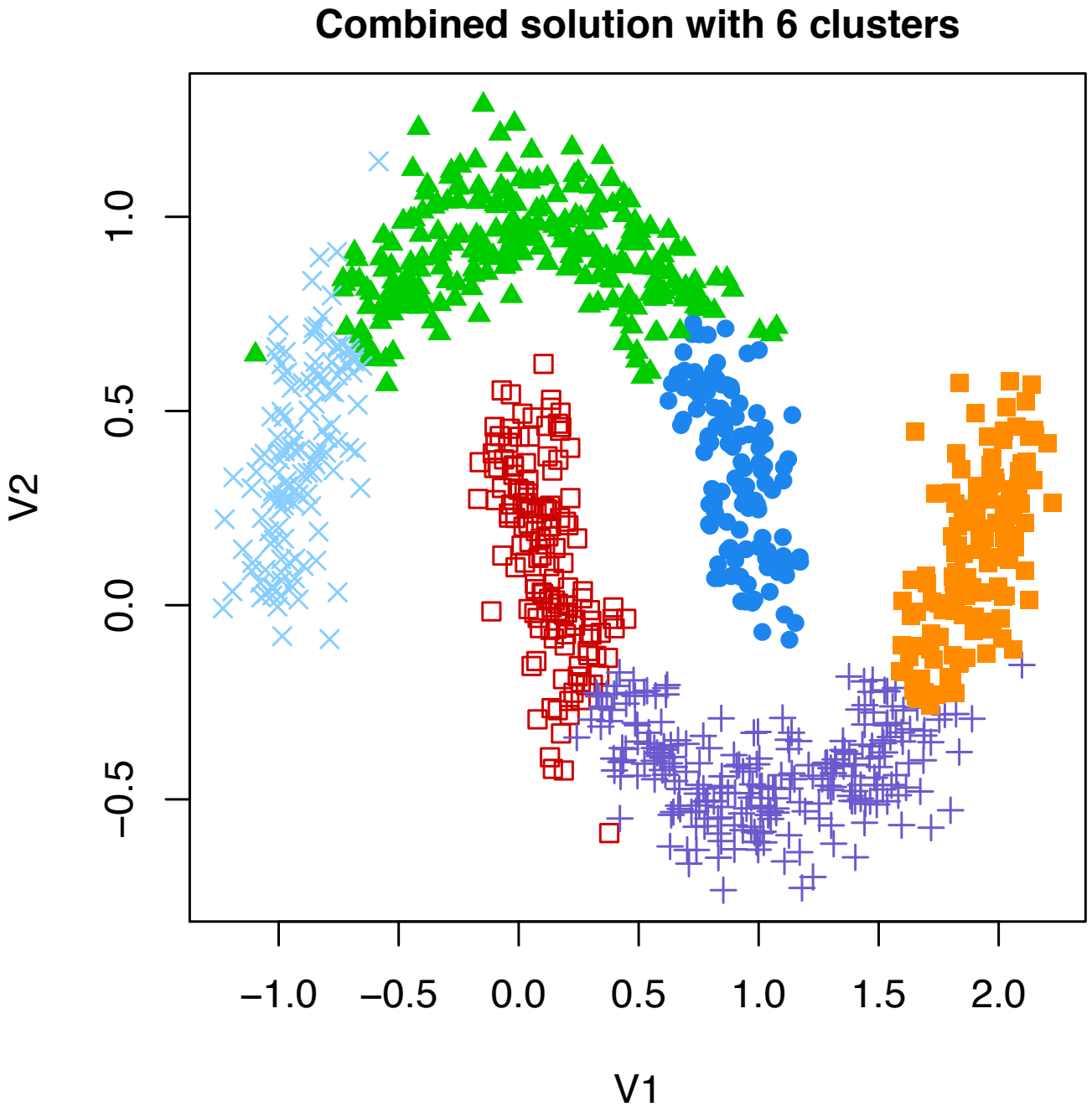
```
library(mclust)

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

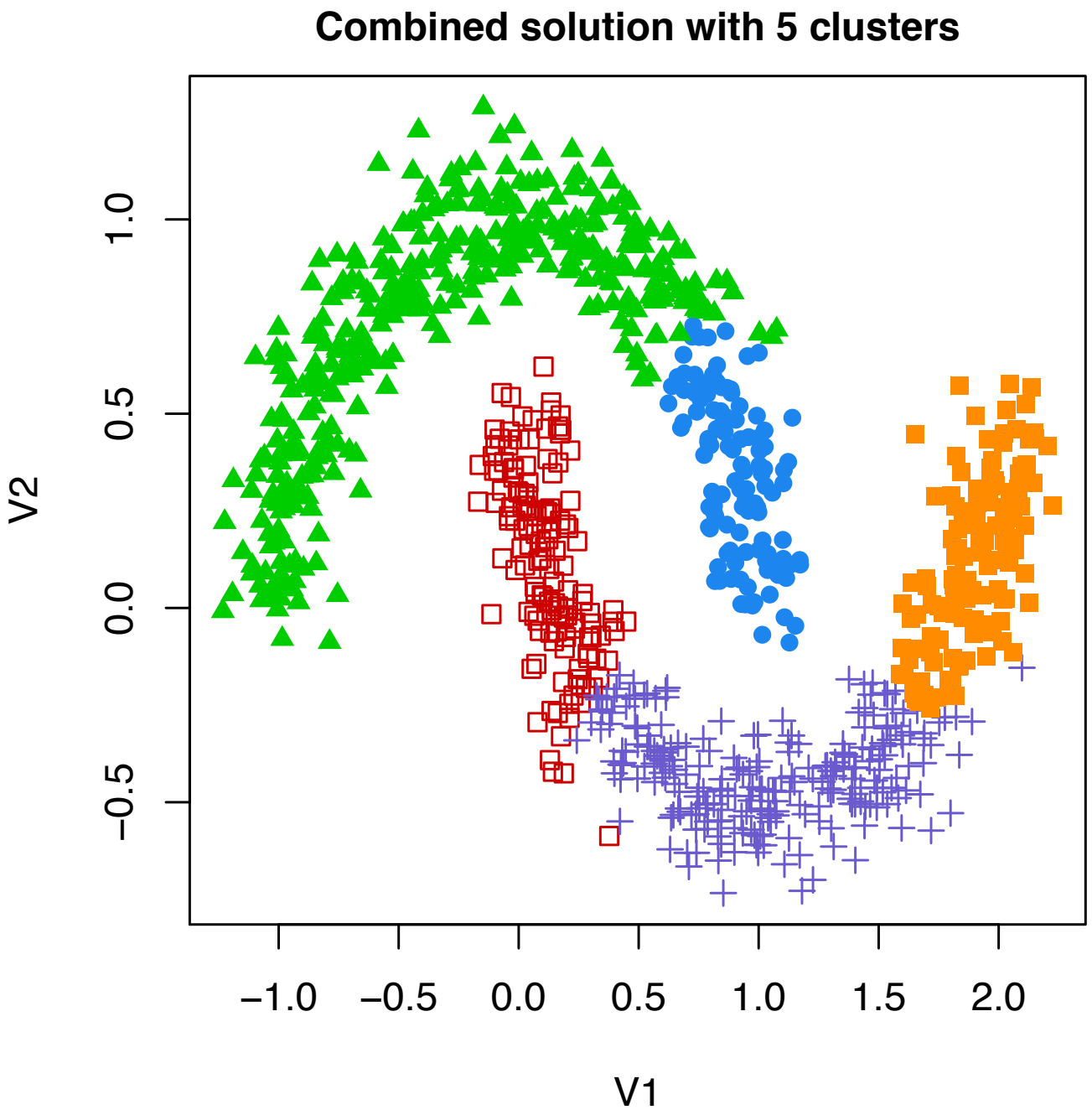
merging with most overlap of the clusters

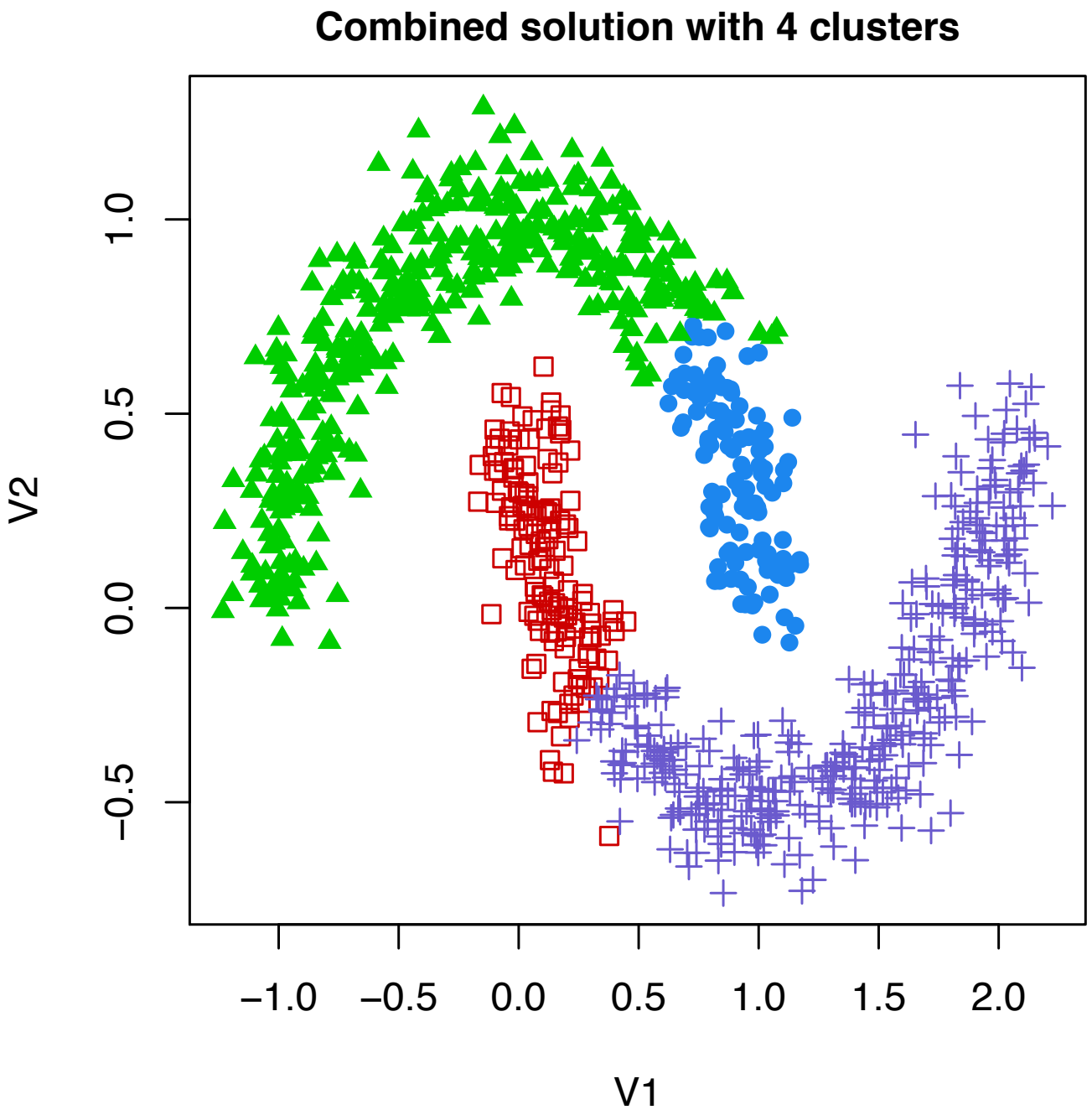


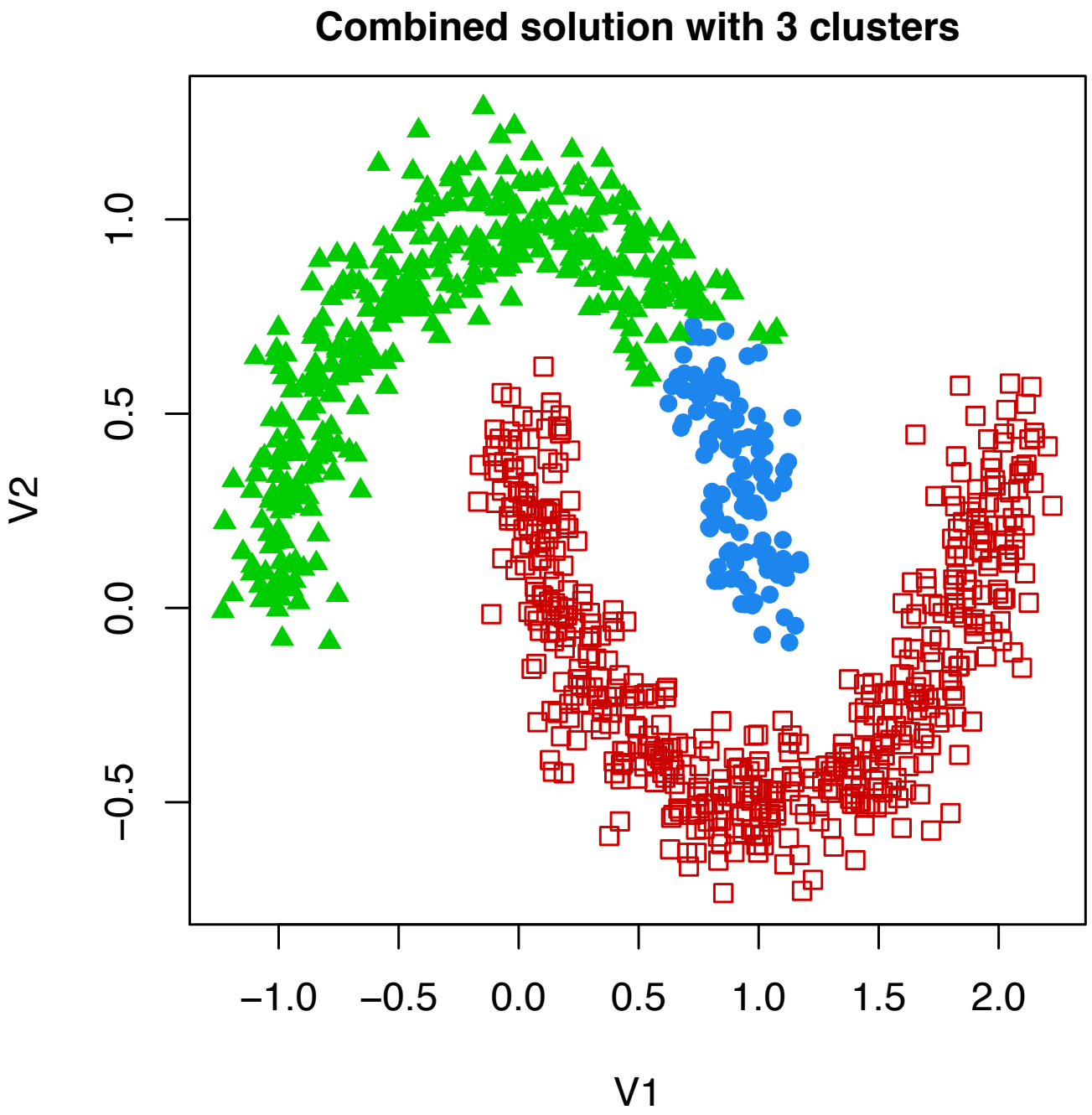

BIC solution (8 clusters)
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1

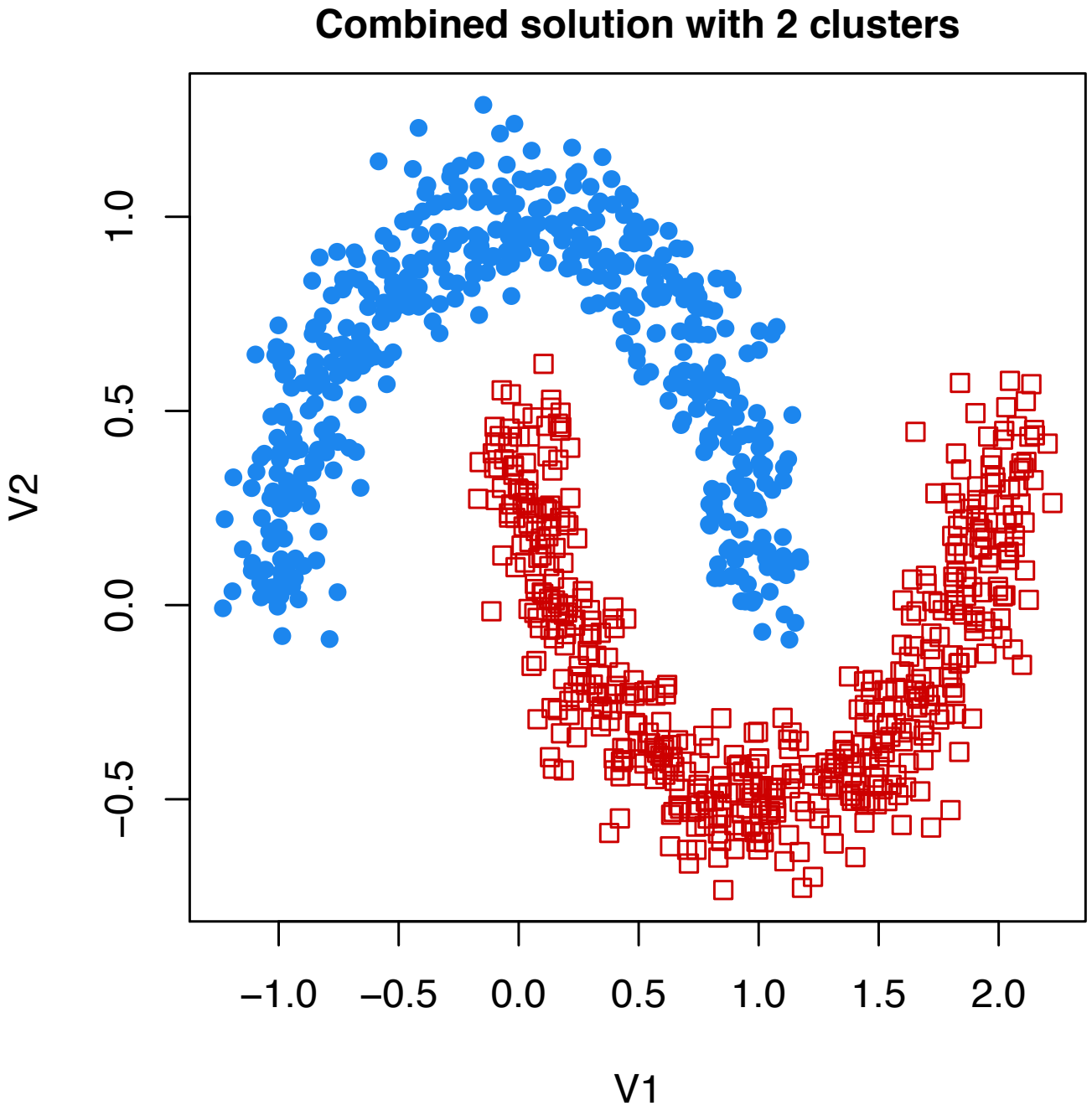

Combined solution with 7 clusters
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1


Combined solution with 6 clusters
V2
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 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

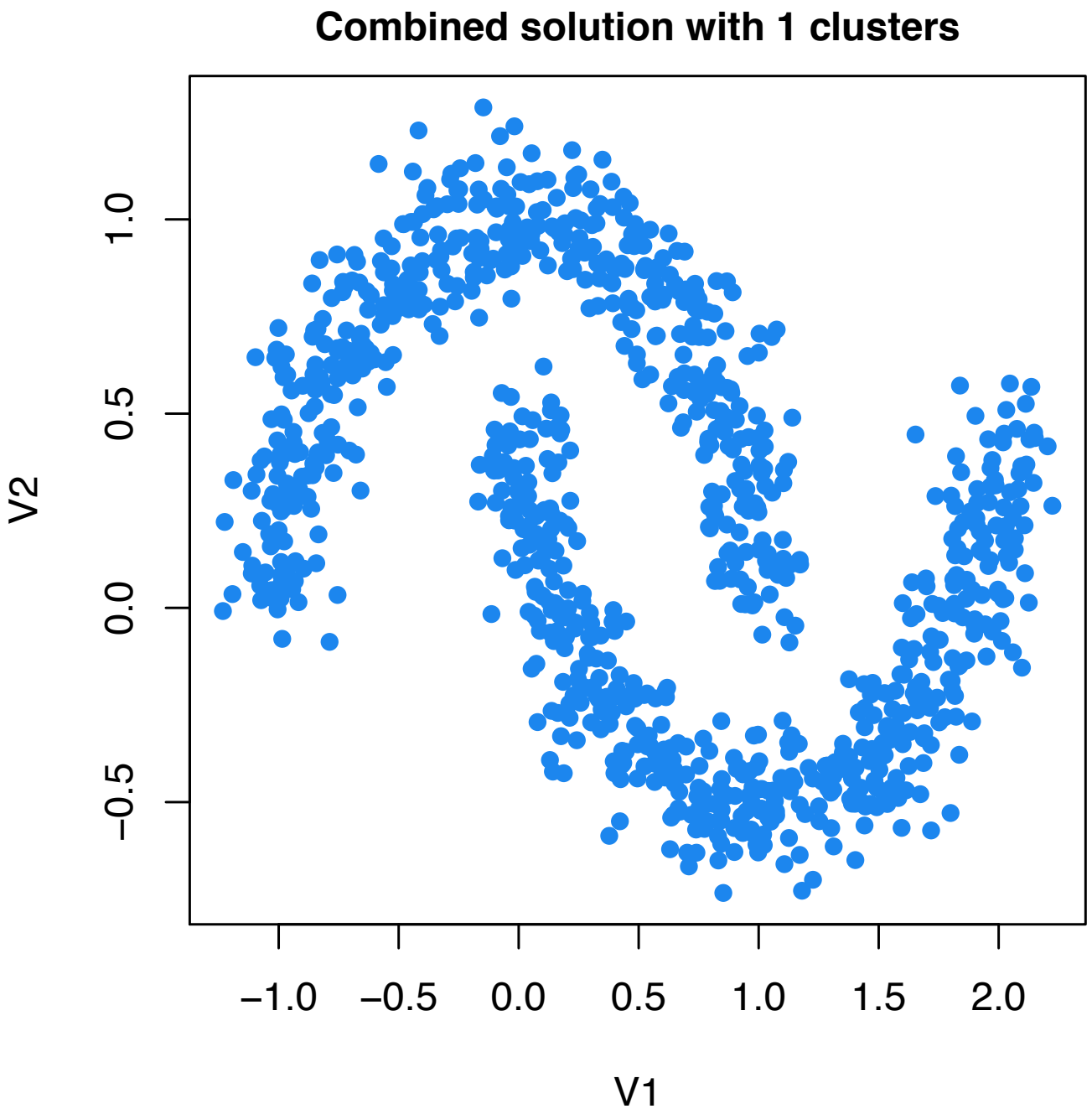

Combined solution with 4 clusters
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1

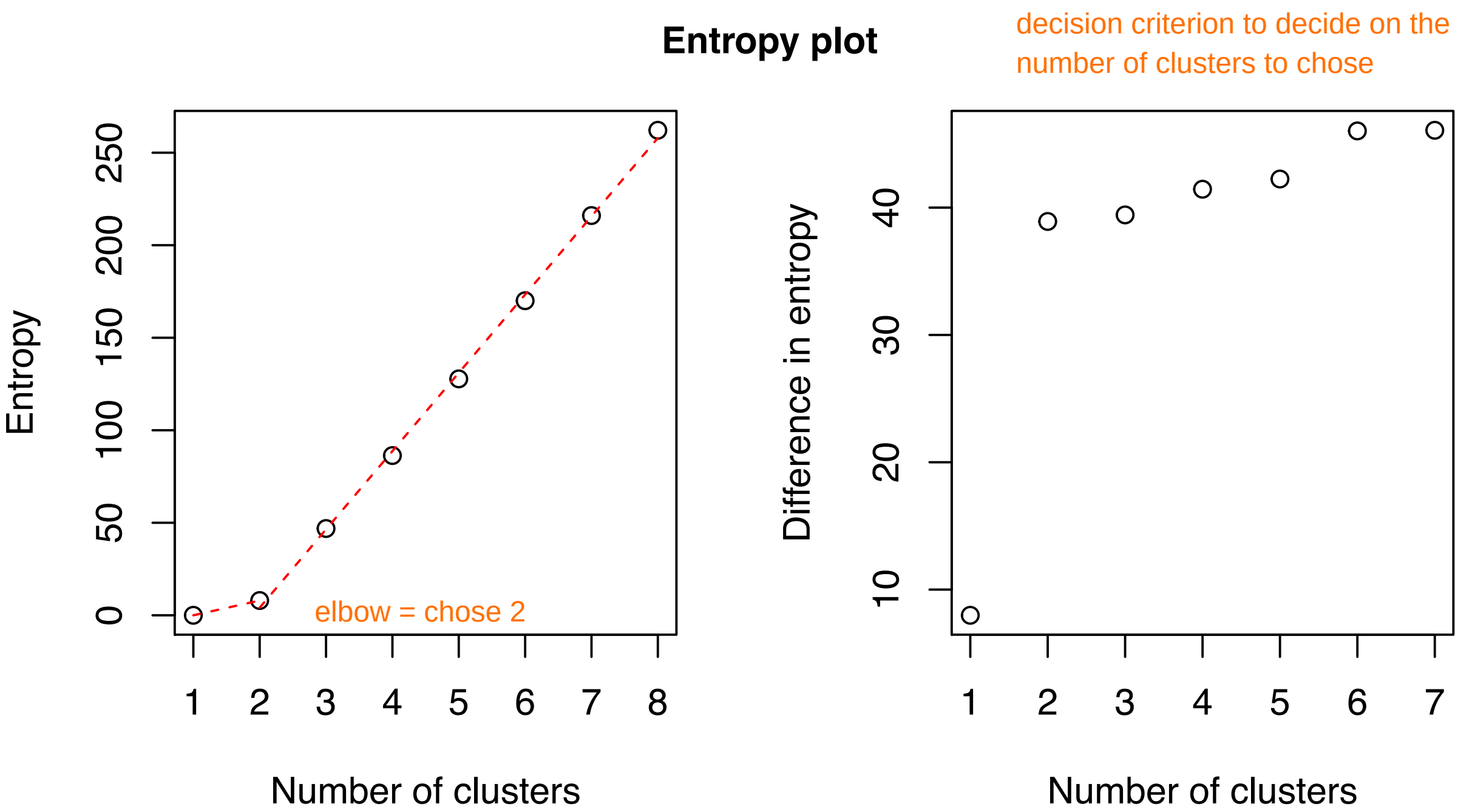

Combined solution with 3 clusters
V2
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
V1



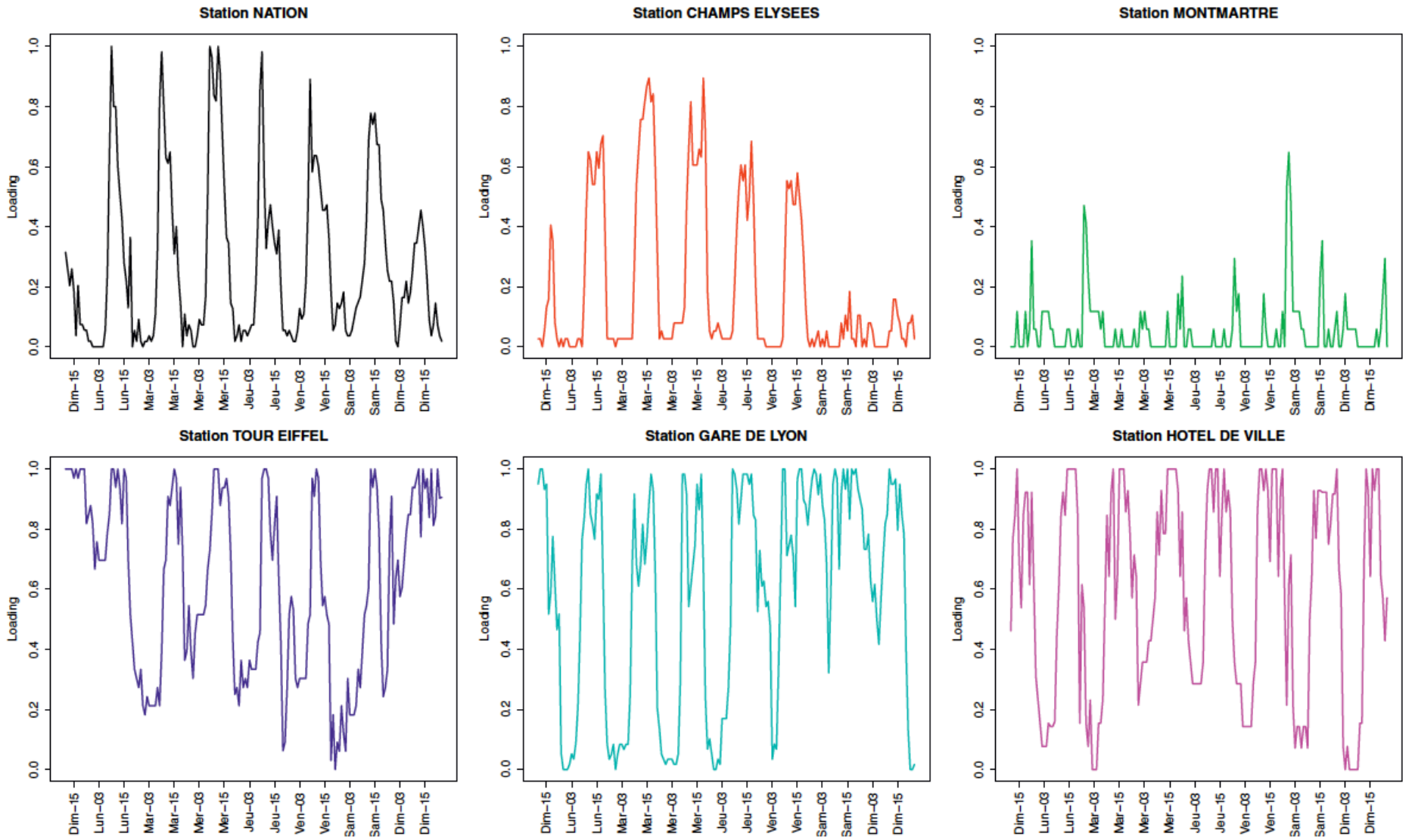
merged till there's only two clusters
- stable solution

very powerful method outperforming a lot more complex methods


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
decision criterion to decide on the number of clusters to chose
Entropy
0
50
100
150
200
250
elbow = chose 2
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



working with time series

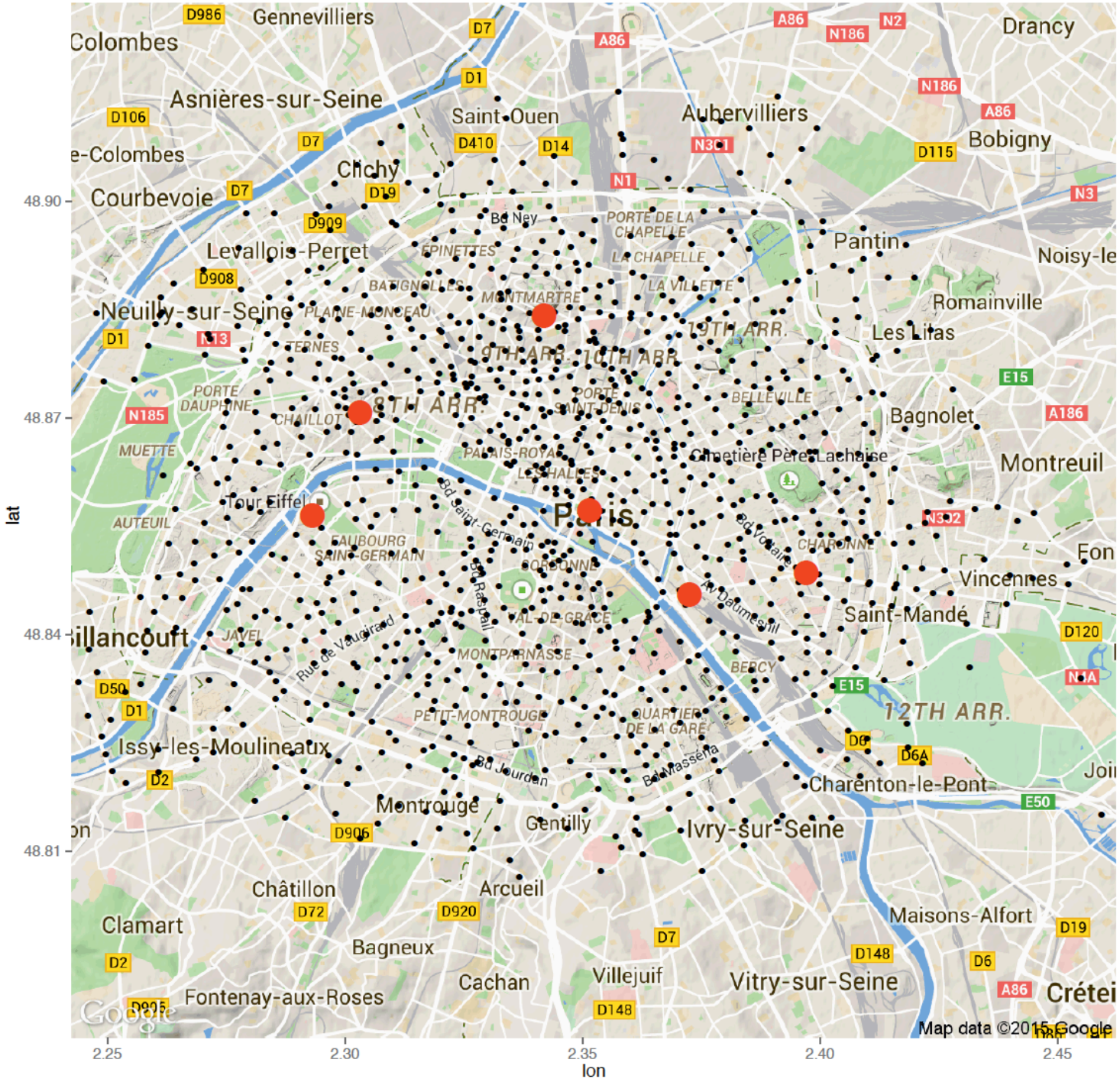
time sharing data

0 = no bike available
1,0 = can't leave the bike at the station

shows that some stations go from 0 to 1 very regularly whereas others aren't used as much

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



Utrecht University

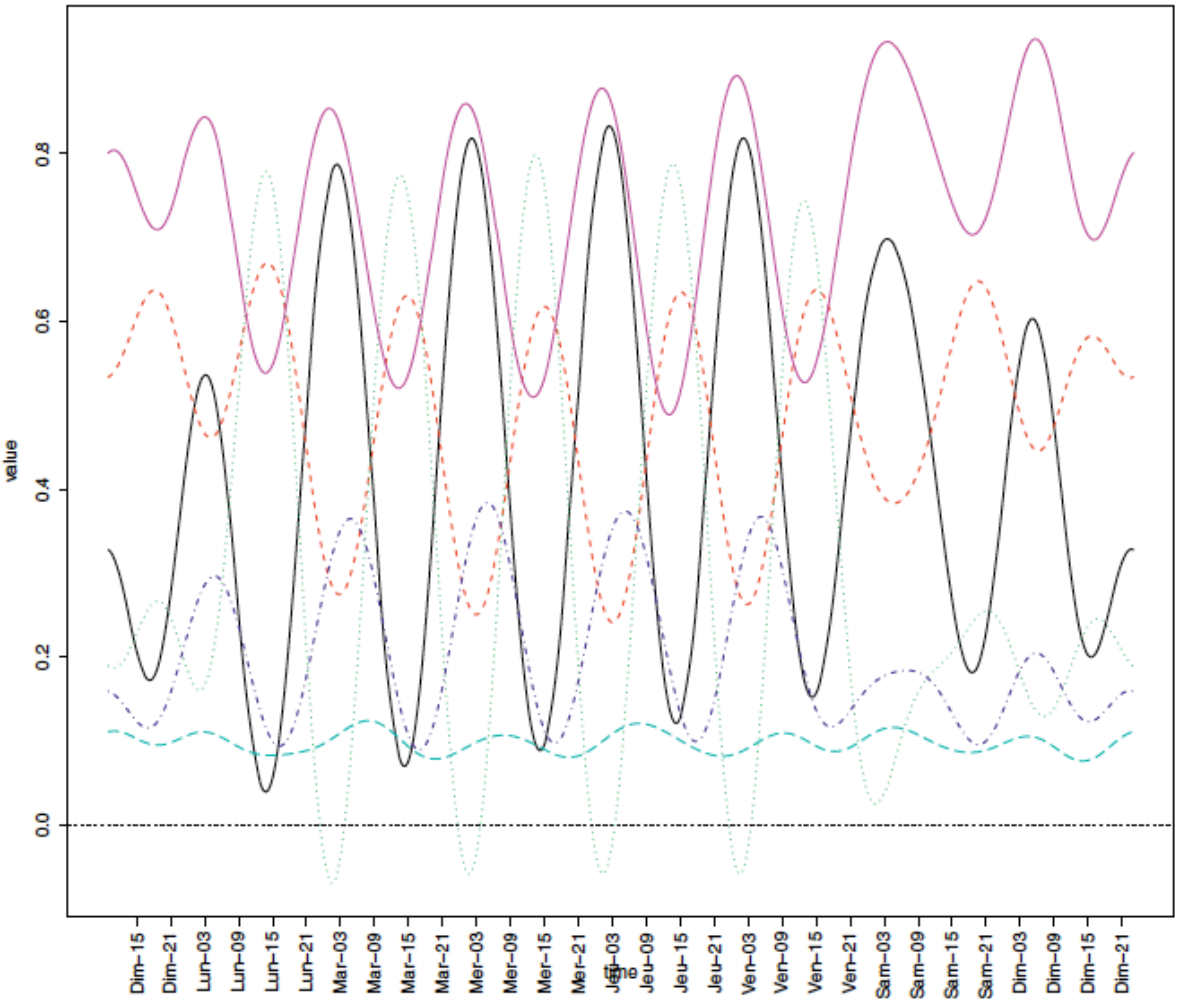
“functional data” clustering in R

```
# Loading libraries and data
library(funFEM)
data(velib)
```

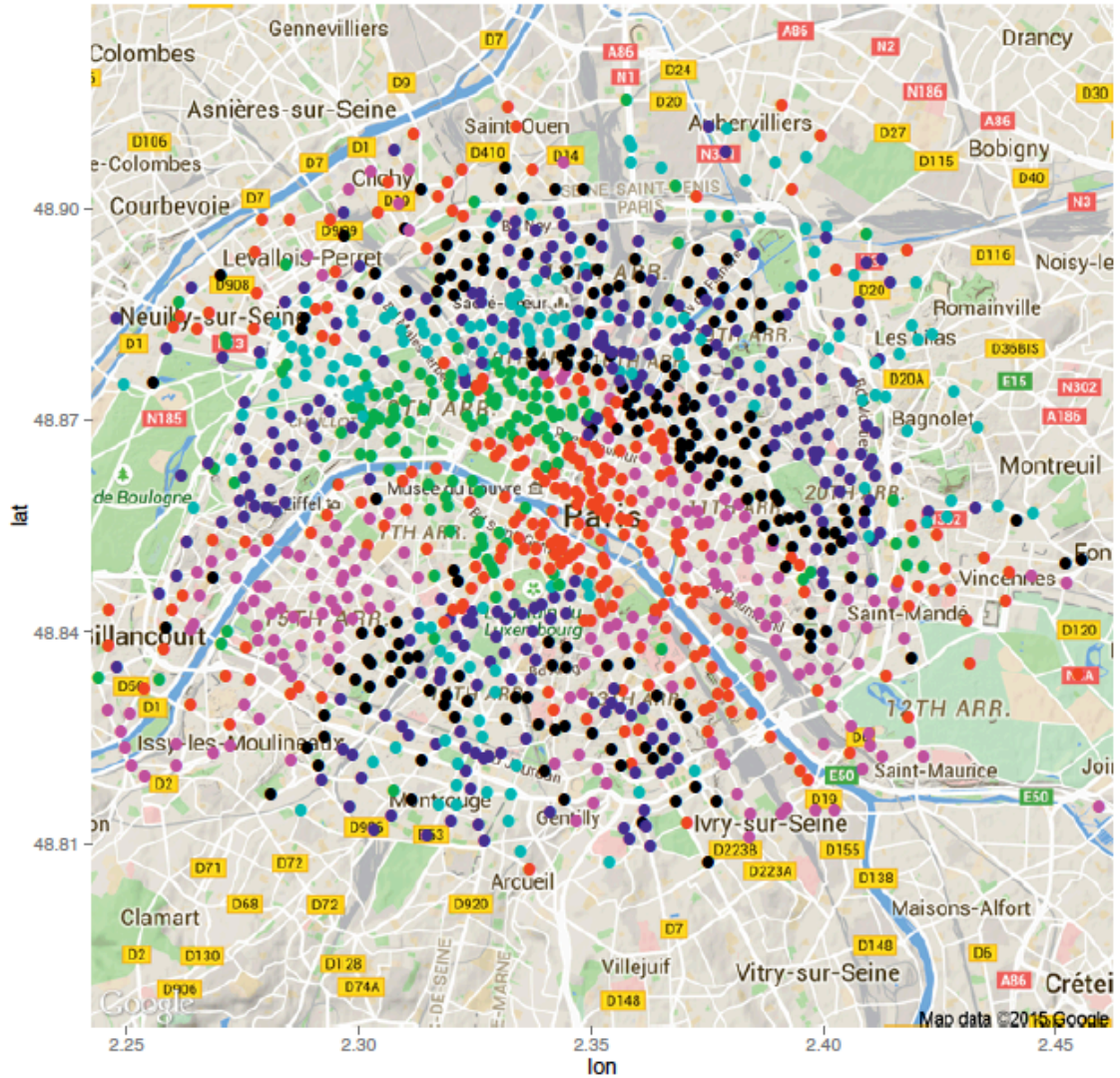
clustering procedure for that type of data

turns the time series data into features where each feature shows the time series differences/distances

```
# Transformation of the raw data as curves
basis = create.fourier.basis(c(0, 181) , nbasis =25)
fdobj = smooth.basis (1:181 ,t(velib$data),basis)$fd


# Clustering with funFEM
res = funFEM(fdobj ,K=6)
```



(a) Cluster mean functions



(b) Map of clustered stations

more important than continuous variables? a lot one can do with categorial data and a lot of info to gain

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

Conclusion

models used to handle NAs- model to make good guesses for the missing values
- EM algorithm
- can be applied when all values in column are missing or when it's just partly missing values

- Model-based clustering: assume “latent classes” that explain why observations cluster assume that observations and their distributions are due to latent classes
- Formalizes assumptions of clusters
- More flexible in its parameterization
- Mixture models can be estimated using the EM algorithm
- Parameters can be tuned using information criteria

trade-off: improving loss and increasing complexity
BIC vs ICL - oftentimes the optimum would have a large amount of components that can/should be merged