Truecluster Examples

Dr. Jens Oehlschlägel

December 2004

Abstract

This file shows the cluster solutions automatically found by the truecluster algorithm for ten example datasets. You will better understand the meaning of the results, if you read the rest of this abstract before jumping to the examples. The example datasets and truecluster solutions are available as comma seperated files at `www.truecluster.com`).

The 10 example datasets are challenging because they contain patterns matching very different *conceptual cluster definitions*. Cluster definitions can be classified into two major groups: *parametric* definitions (looking for specific shapes) and *non-parametric* definitions, which allow for more general shapes but require stronger differences in data densities, e.g. well-seperated clusters. Specific cluster algorithms do imply *technical cluster definitions* which belong to the parameteric or non-parametric classes as well. Examples for the parametric class are algorithms like "KMEANS", "Partitioning around Medoids (PAM)", "Model based Clustering" (S+) and "Two-Step Clustering" (SPSS). Examples for the non-parametric class are "single link agglomeration (SLA)", "DBSCAN" or "MODECLUS" (SAS). *All* of these technical cluster definitions require important *detail choices* of the user - sometimes implicitely by accepting the defaults - before the software actually generates a *physical cluster solution*, i.e. actually assigns cases to a finite number of clusters. The final solution may or may not represent patterns present in the data, due to possibly wrong choice of technical cluster definition and possibly wrong choice of algorithmic details. By constructing examples in which data patterns correspond to conflicting conceptual cluster definitions, it is easy to prove that objective choice of technical cluster definition is not always possible. However, by constructing a common statistical framework considering the match between data densities and cluster solutions, it is possible to objectively optimize the choice of the algorithmic details ... and to some degree also the choice of the cluster definition.

Truecluster is a set of rules and algorithms which puts parametric and non-parametric technical cluster definitions into a *common statistical framework* and into a *common technical framework*. The main benefit of the common technical framework is *scalability* and *optimality*: not all technical cluster definitions are scalable to arbitrary big data sets (e.g. single link agglomeration) and not all technical cluster definitions grant convergence to an optimal solution given the detail choices (e.g. KMEANS). Truecluster turns very different technical cluster definitions into scalable and optimal algorithms. The main benefit of the common statistical framework is optimization of the detail choices, e.g. choosing the correct number of clusters for KMEANS. For any technical cluster definition matching the data pattern, truecluster will choose optimal details and thus an optimal physical cluster solution. We have applied truecluster versions of four cluster definitions: KMEANS and PAM as examples of parametric definitions and SLA and DBSCAN as examples of non-parametric definitions.

While checking the excellent truecluster results on the 10 example datasets you should try to answer the following two questions: 1) Does the technical cluster definition match the data pattern? 2) Does the physical cluster solution - especially the number of clusters - match the data pattern? The following table gives an overview on cluster definitions and example datasets. All charts have a parametric definition on the left side and a non-parametric on the right side.

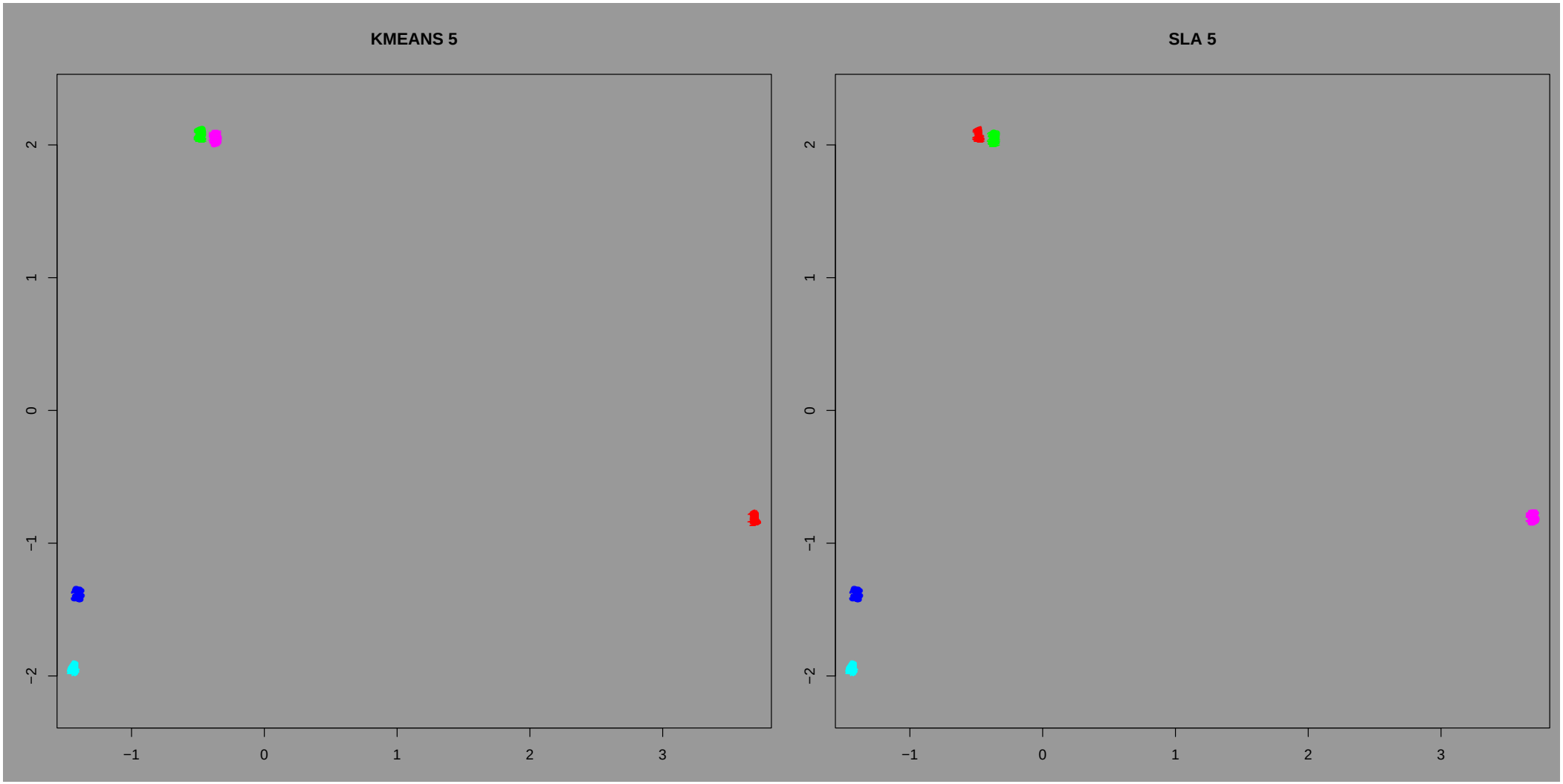
Table 1: Overview

	KMEANS: Looks for equally sized spherical clusters. Detail choice: no. of clusters.	**PAM:** Looks for rather equally sized convex clusters. Detail choice: no. of clusters.	**SLA:** Grows clusters by recursively joining those cases or clusters with nearest neighbors. Detail choice: no. of clusters.	**DBSCAN:** Grows clusters of min. size by recursively adding cases within a max. distance (cases may remain unclassfied as outliers). Detail choice: maximum distance (at min. size 5).
Cophenetic: 5 well separated equally sized spherical clusters with possibility for hierarchical subgrouping	5 OK	5 OK	5 OK	5 OK
Noflipper: 4 well separated equally sized spherical clusters in 2 unambiguous subgroups	4 OK	4 OK	2 OK	2 OK
Flipper: 4 well separated equally sized spherical clusters with conflicting subgrouping possibilities	4 OK	4 OK	4 OK	4 OK
Symflipper: 4 equally sized spherical clusters wich are not well seperated	4 OK	4 OK	2 (expected failure)	2 (expected failure)
SASmodeclus: 3 well separated clusters of different shape and orientation	4 (expected failure)	3 (only one misclassification)	3 OK	3 OK
Extremegroup: 3 well separated unequally sized spherical clusters	5 (expected failure)	3 OK	3 OK	3 OK
Hierarchic: 3 well separated unequally sized spherical clusters, with the third one subpartitioned the same way (5 in total)	4 (expected failure)	6 (expected failure)	3 OK (but not found 5, due to global distance definition)	3 OK (but not found 5, due to global distance definition)
Conflict: 2 well separated elongated clusters	7 (expected failure)	6 (expected failure)	2 OK	2 OK
CenterRing: 2 well separated clusters, one topologically enclosing the other	10 (expected failure)	10 (expected failure)	2 OK	2 OK
Spiral: 2 well separated clusters, both enclosing each other	4 (expected failure)	8 (expected failure)	2 OK	2 OK

List of Figures

Figure 1: Cophenetic

We start with a simple example of 5 equally sized clusters forming a natural hierarchic grouping. Using all four cluster definitions, truecluster finds exactly those 5 clusters. As the cluster are very small in relation to the distances between clusters, the plotted numbers showing cluster membership are visually overlapping. All cases are perfectly assigned.



Cophenetic continued . . .

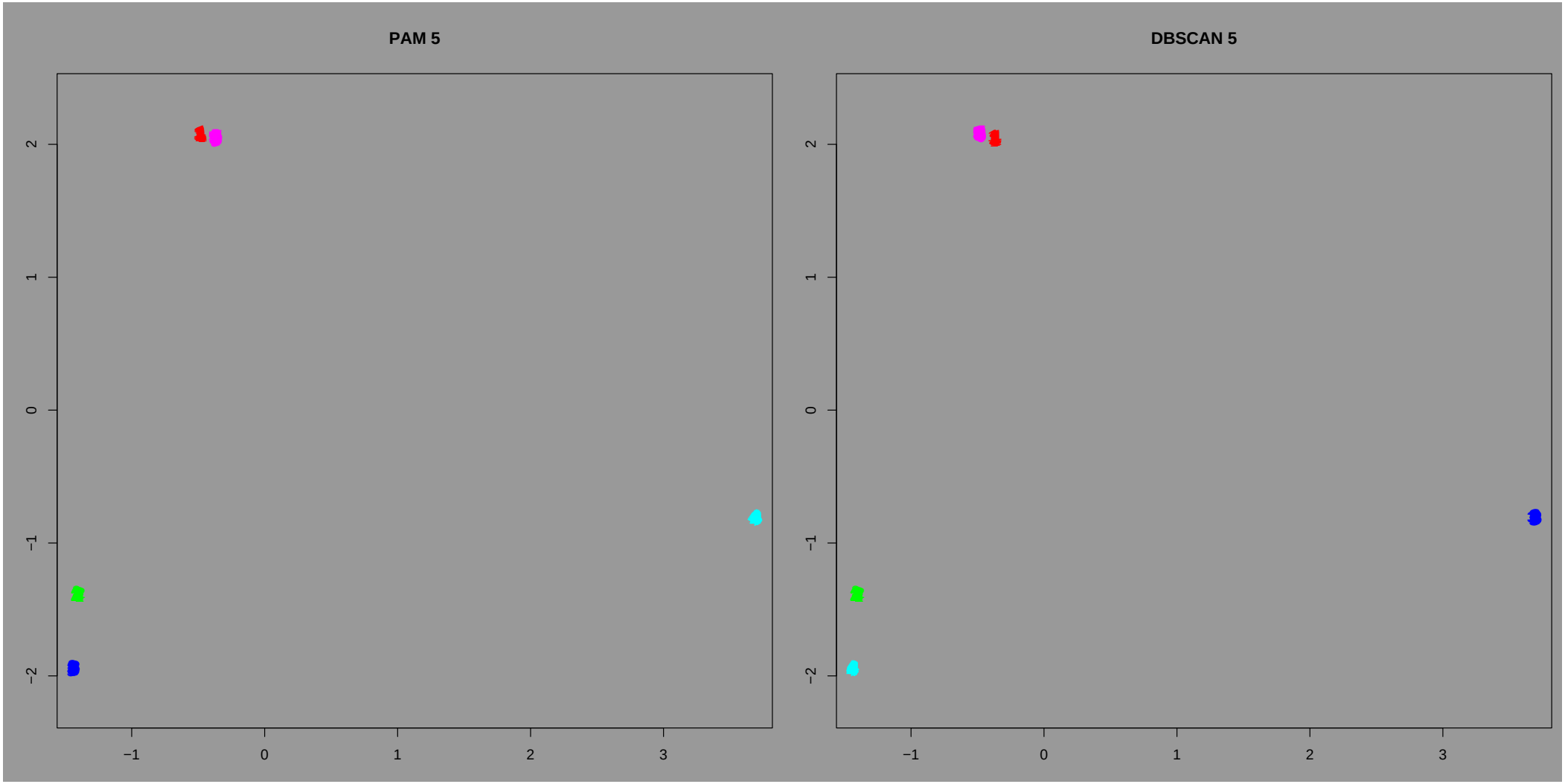
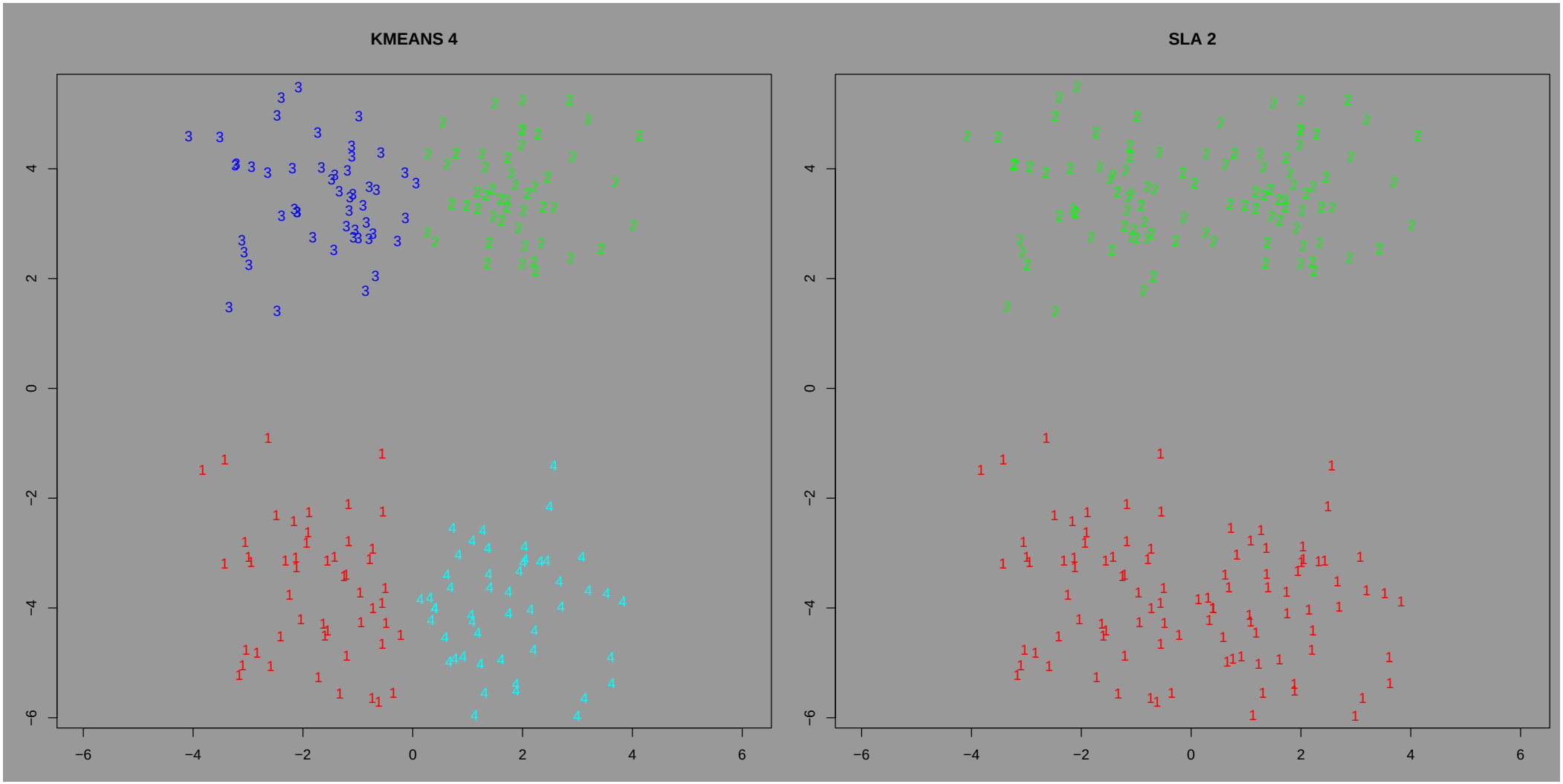


Figure 2: Noflipper

Here we have 4 equally sized spherical clusters which can be grouped unambiguously into 2 non-spherical segments. Both of these reasonable solutions are found by truecluster, the 4 cluster solution by the parametric cluster definitions KMEANS and PAM and the 2 cluster solution by the non-parametric SLA and DBSCAN.



Noflipper continued . . .

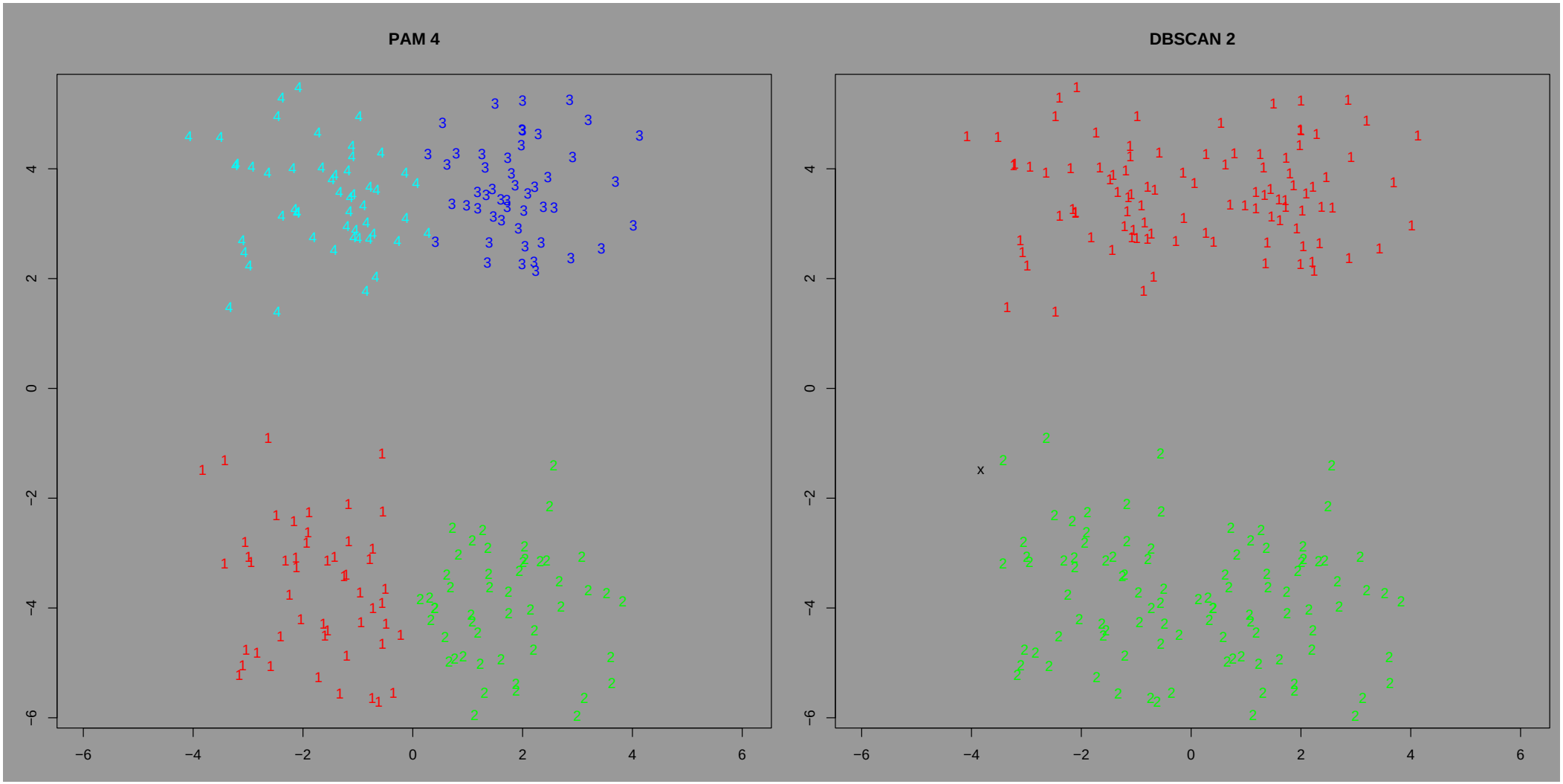
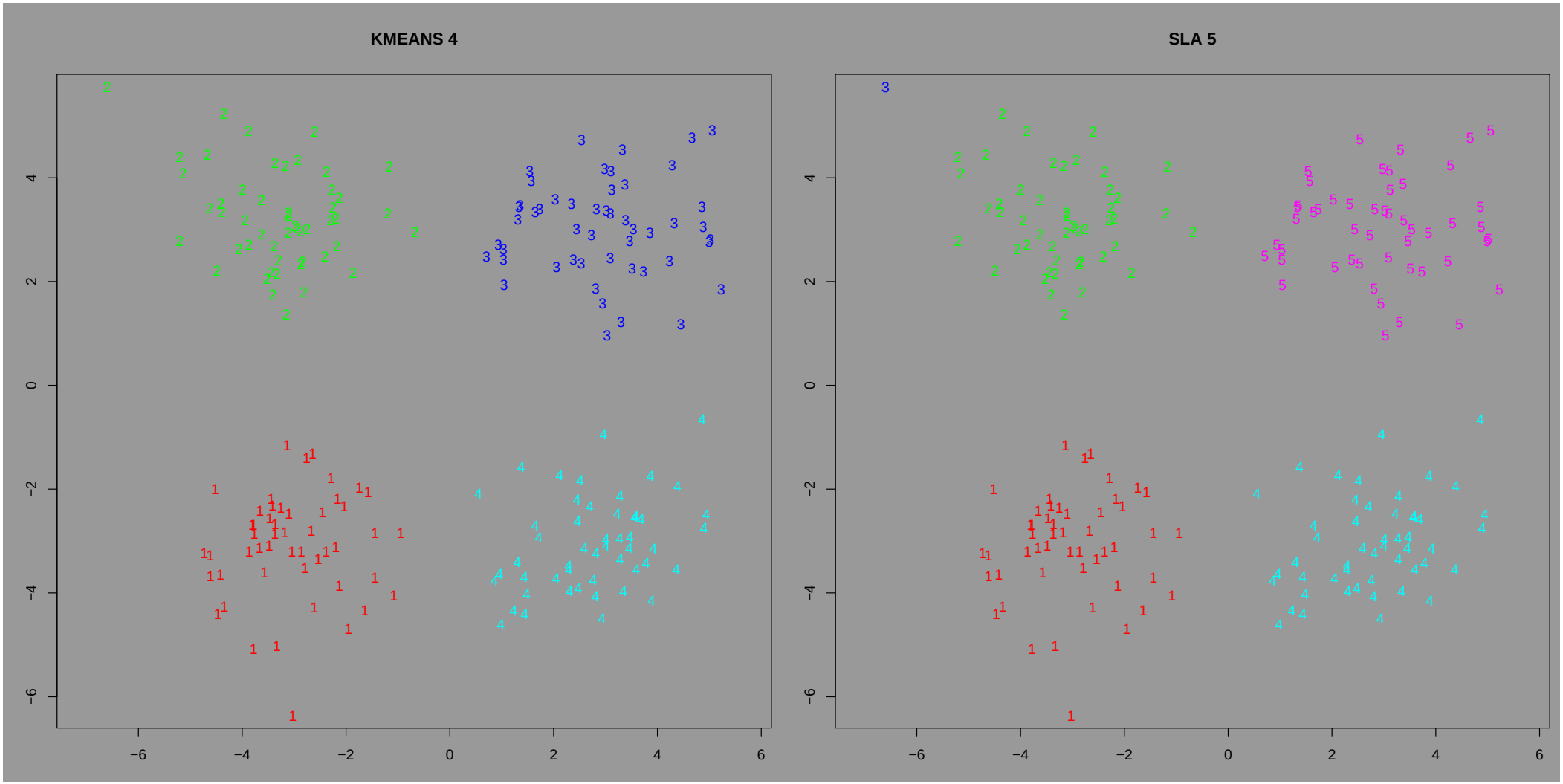


Figure 3: Flipper

Here we have also 4 equally sized spherical cluster. Differing from the previous *Noflipper* example, here are *two* possibilities for grouping (vertical and horizontal). Thus there is no natural 2-cluster solution, and the 4 cluster solution is identified by truecluster whichever of the four cluster definitions is used.



Flipper continued . . .

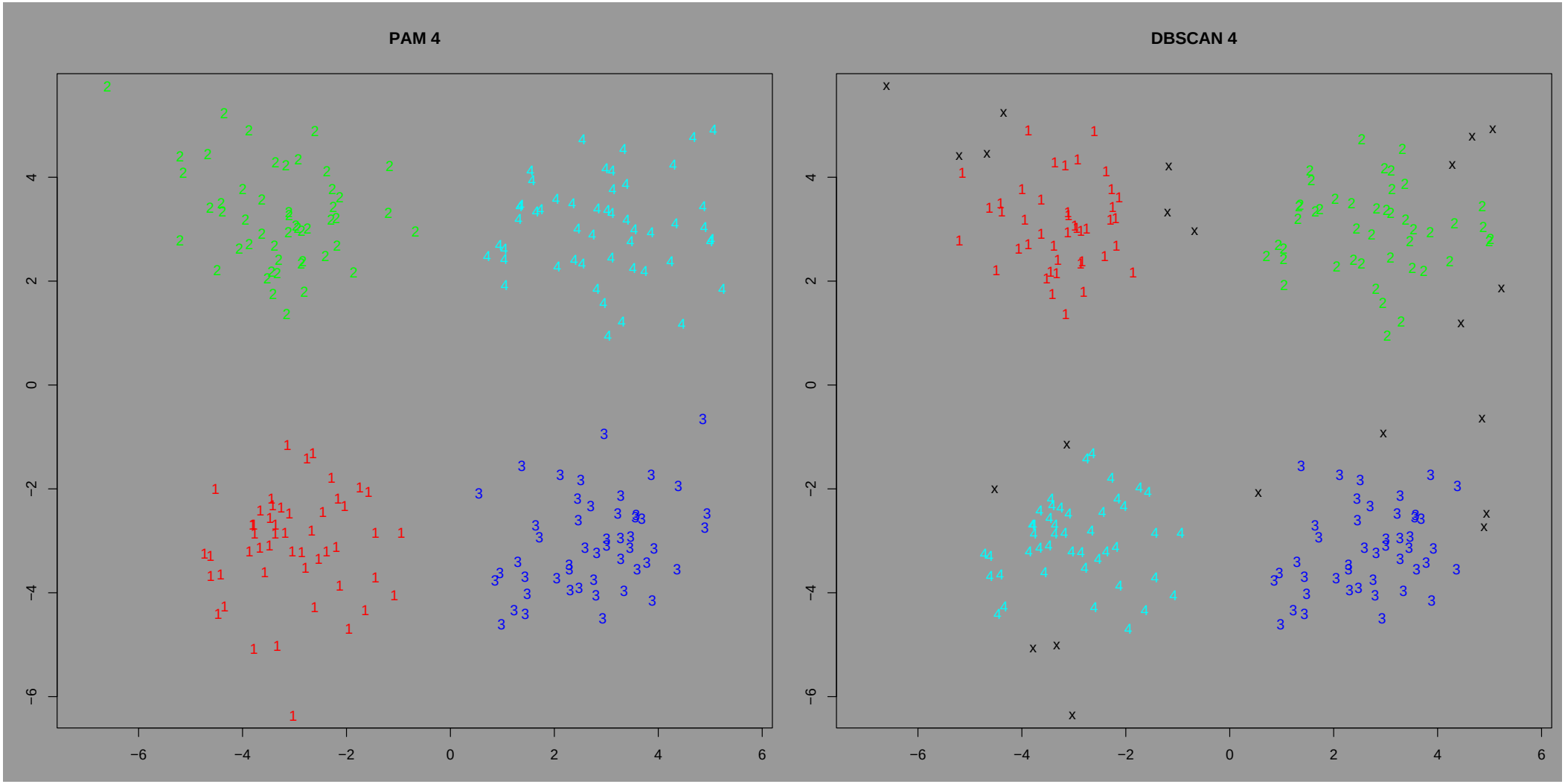
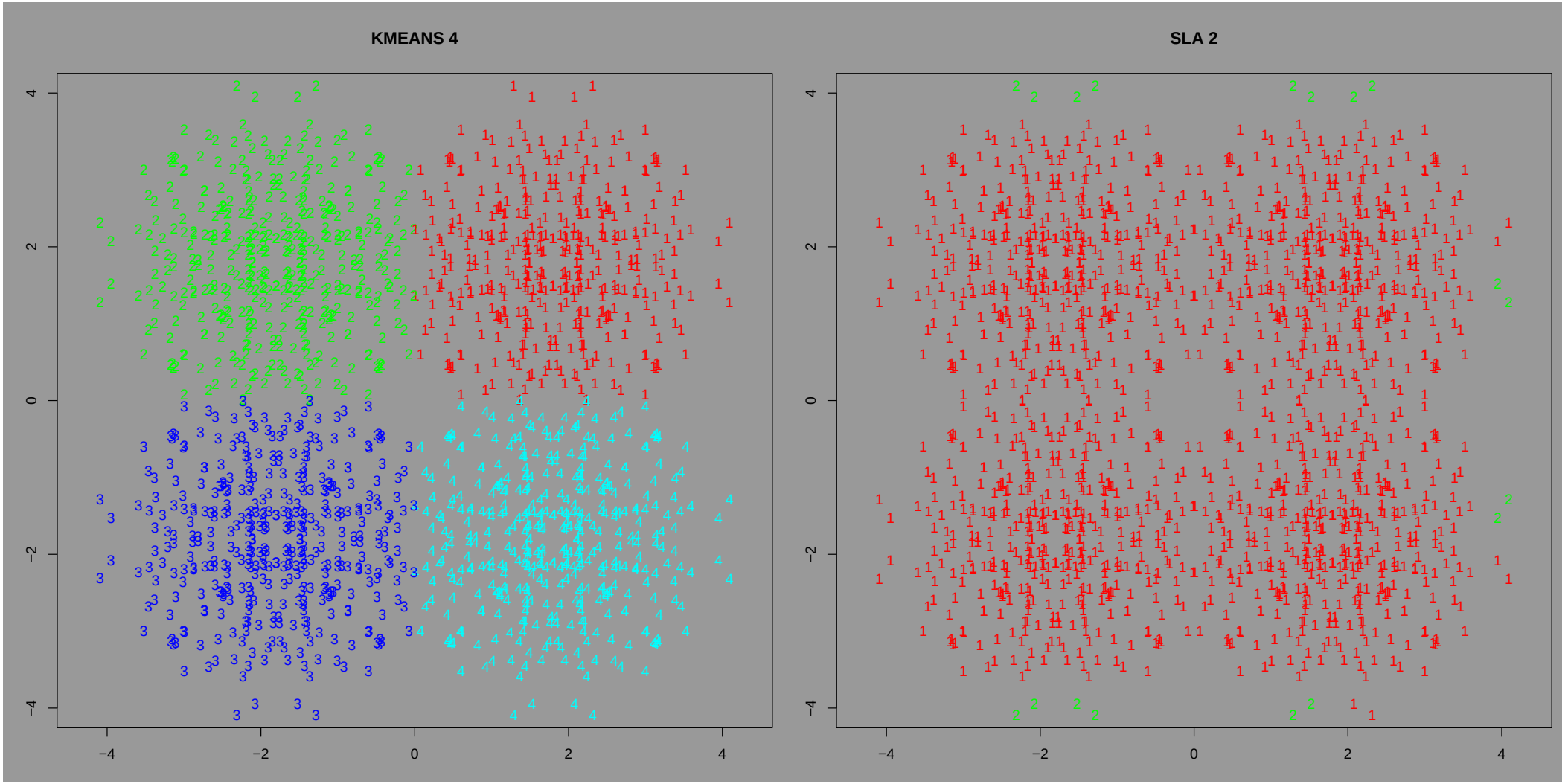


Figure 4: Symflipper

Here the 4 groups are not seperated good enough to be found by a non-parametric method. However, truecluster identifies the 4 clusters under the assumptions of the parametric definitions KMEANS and PAM.



Symflipper continued . . .

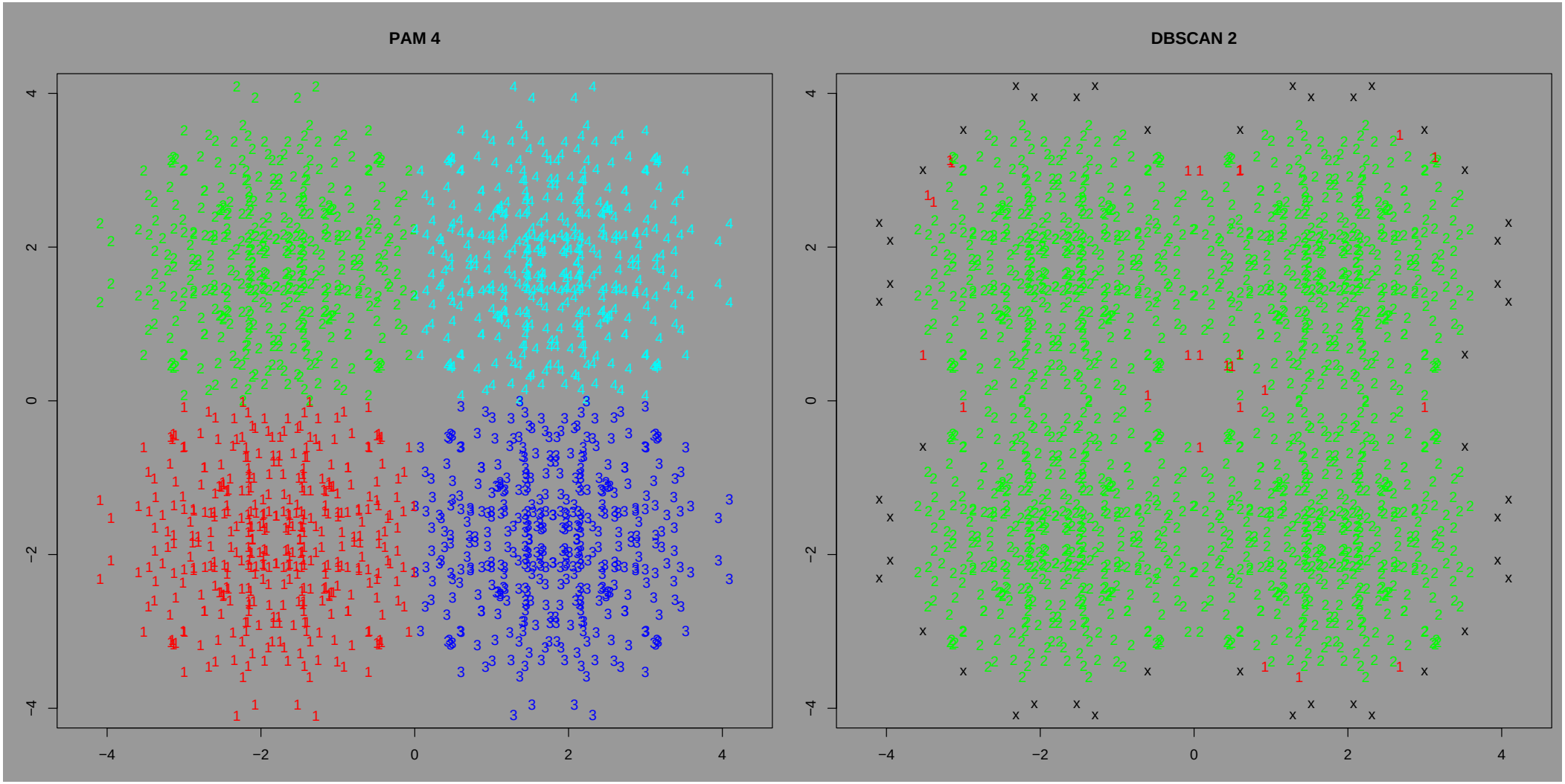
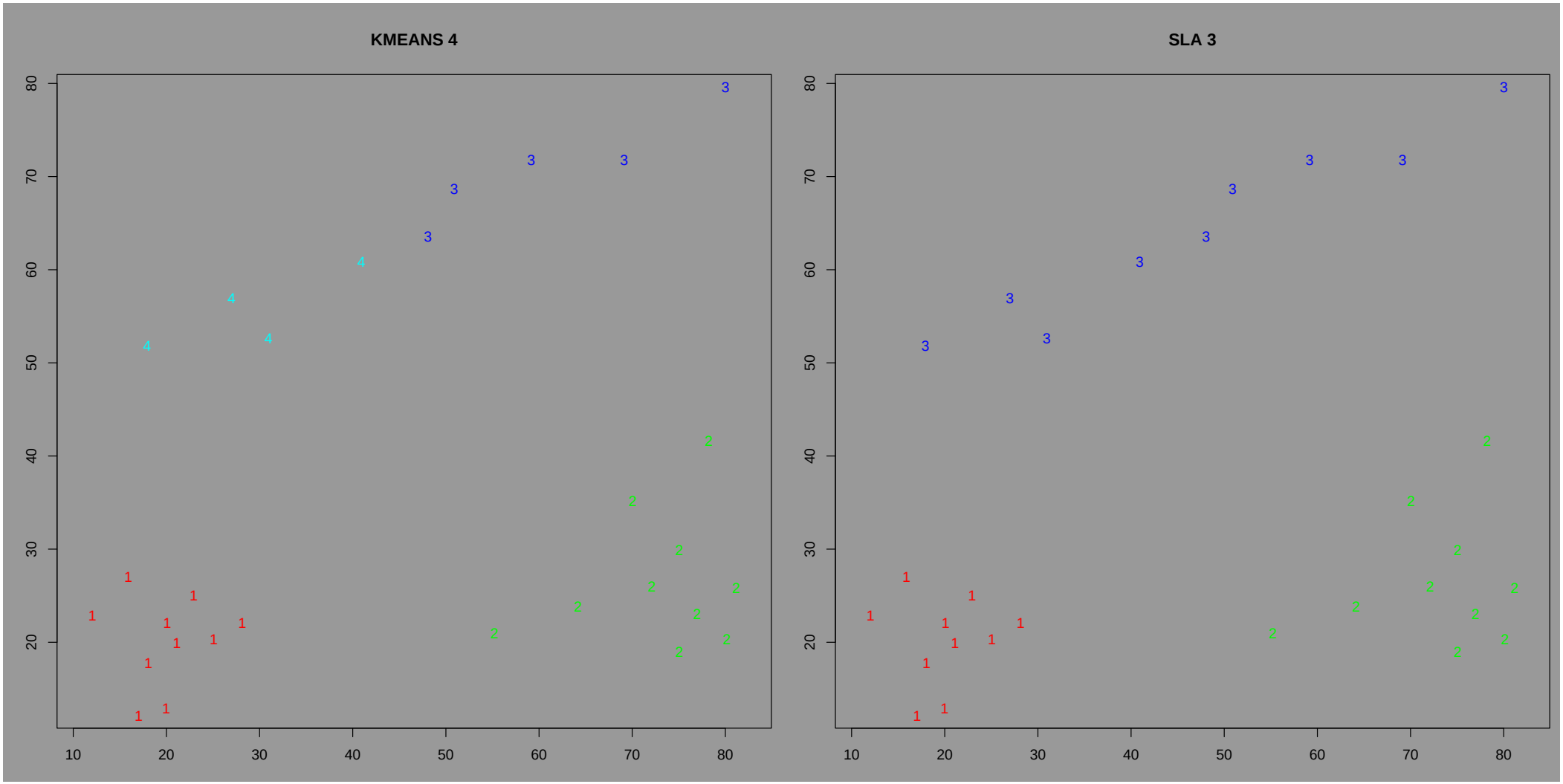


Figure 5: SASmodeclus

This is a 3 cluster configuration presented by SAS as an example of a situation which requires a density based method. SAS MODECLUS will identify the 3 clusters (after tweaking their algorithm's parameters). The truecluster implementations of SLA and DBSCAN automatically find the three clusters without any subjective parameter choices. As expected, KMEANS and PAM fail on this example (PAM almost got it right, but did assign one case to the wrong cluster).



SASmodeclus continued . . .

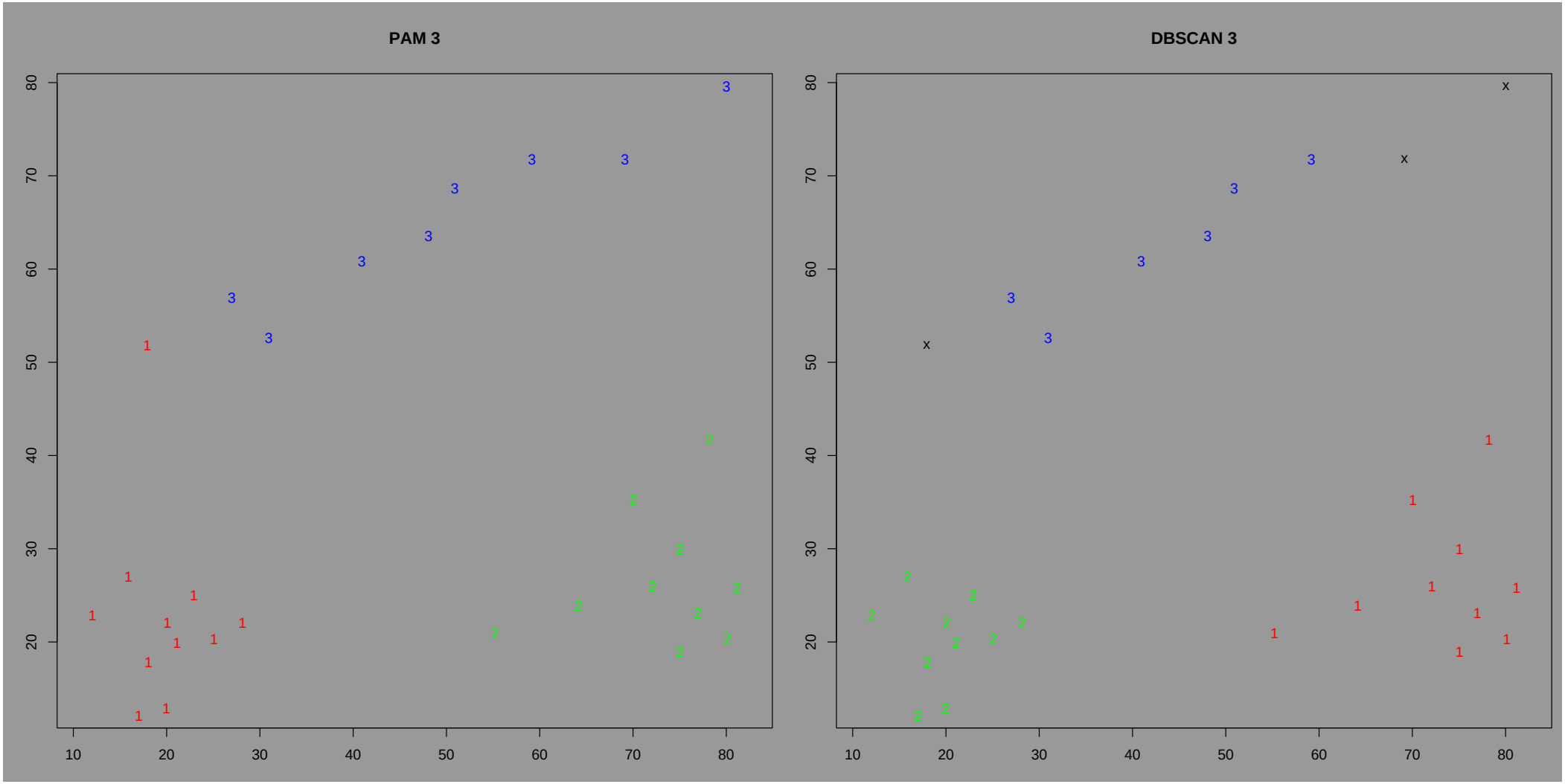
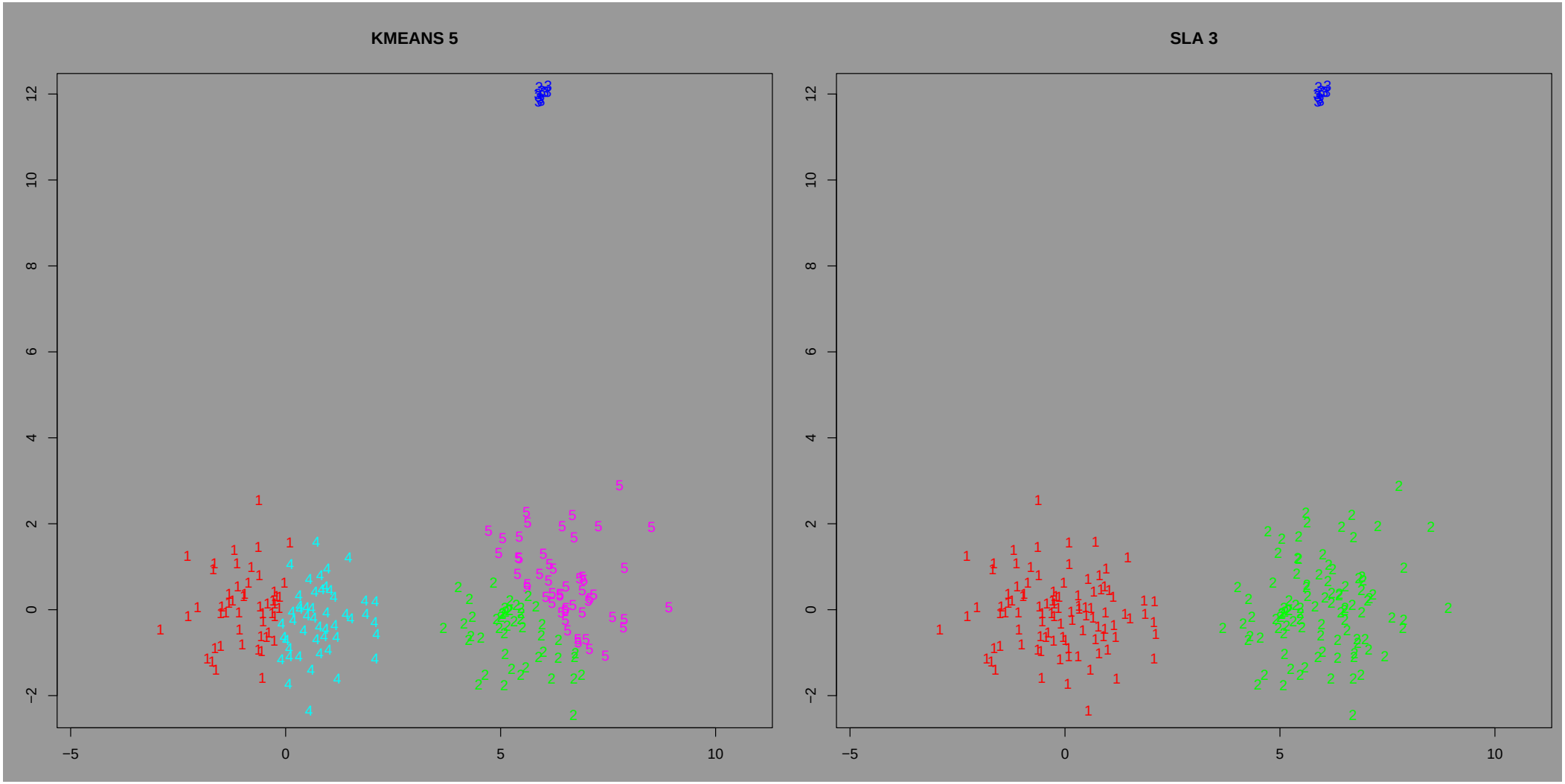


Figure 6: Extremegroup

This is a 3 cluster example with very unequally sized spherical clusters. Since the clusters are well seperated, the truecluster implementations of the non-parametric definitions SLA and DBSCAN easily recognize these 3 clusters. As expected KMEANS fails here, while PAM is more robust and also identifies the 3 clusters.



Extremegroup continued . . .

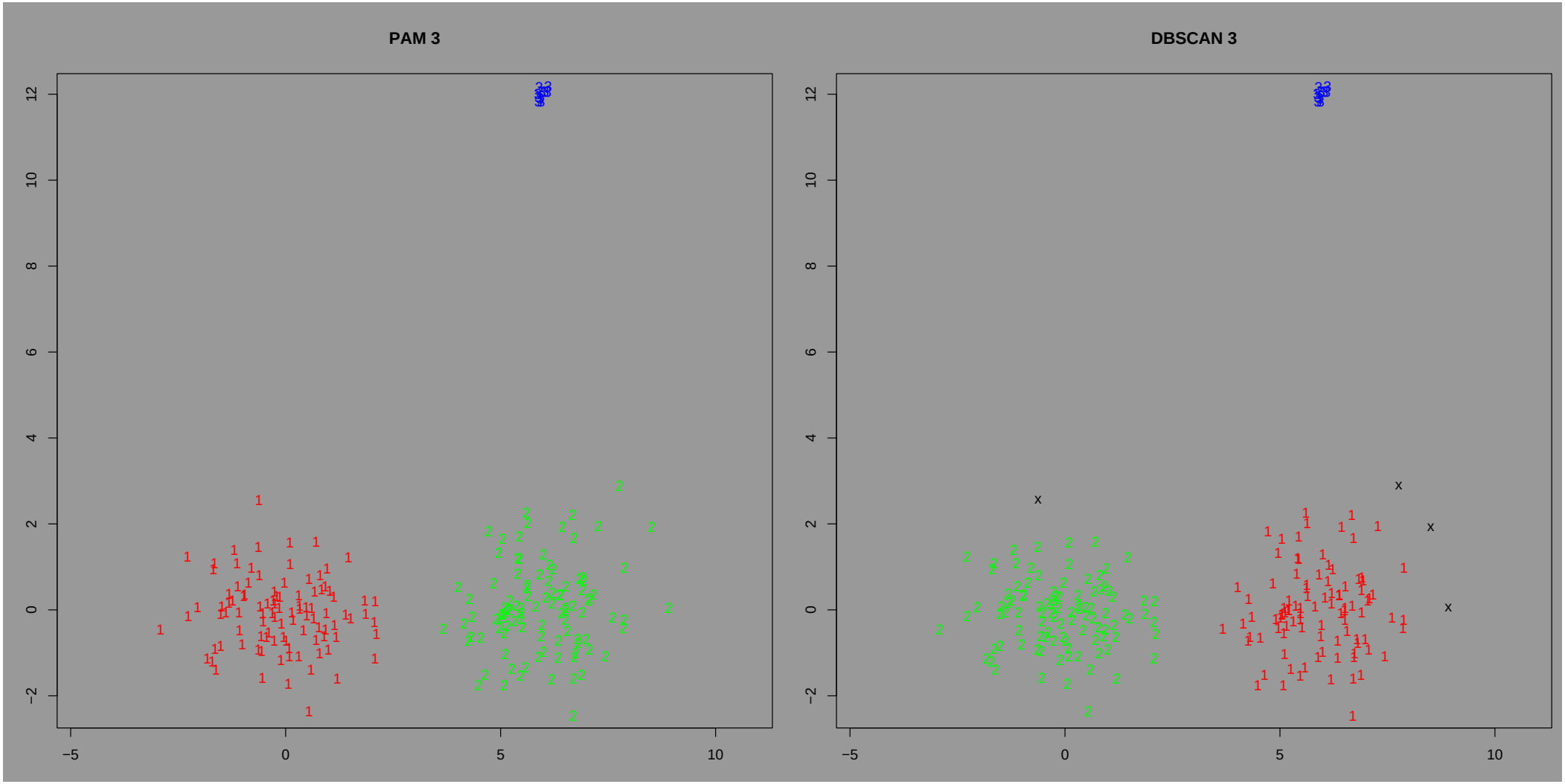
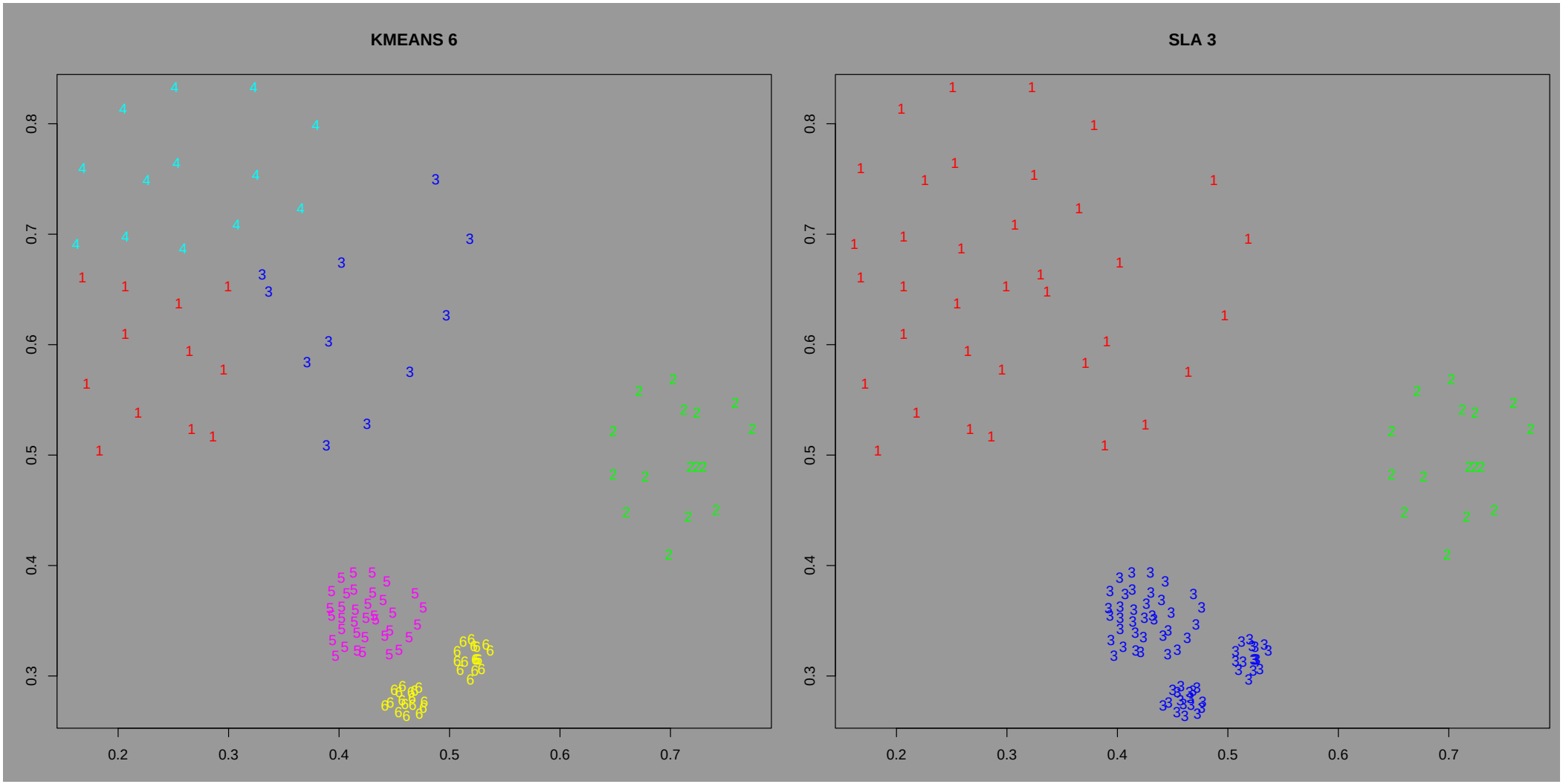


Figure 7: Hierarchic

Here we have 3 well separated unequally sized spherical clusters, with the third one sub-partitioned the same way (5 in total). Both non-parametric defintions SAL and DBSCAN identify the 3 cluster solution correctly. That they 'prefer' the 3 cluster solution over the 5 cluster solution is due to their global handling of 'distance'. Reapplying SAL or DBSCAN to the third cluster would correctly subdivide it. As expected, the parametric definitions do not give good solutions, because their assumptions are violated.



Hierarchic continued ...

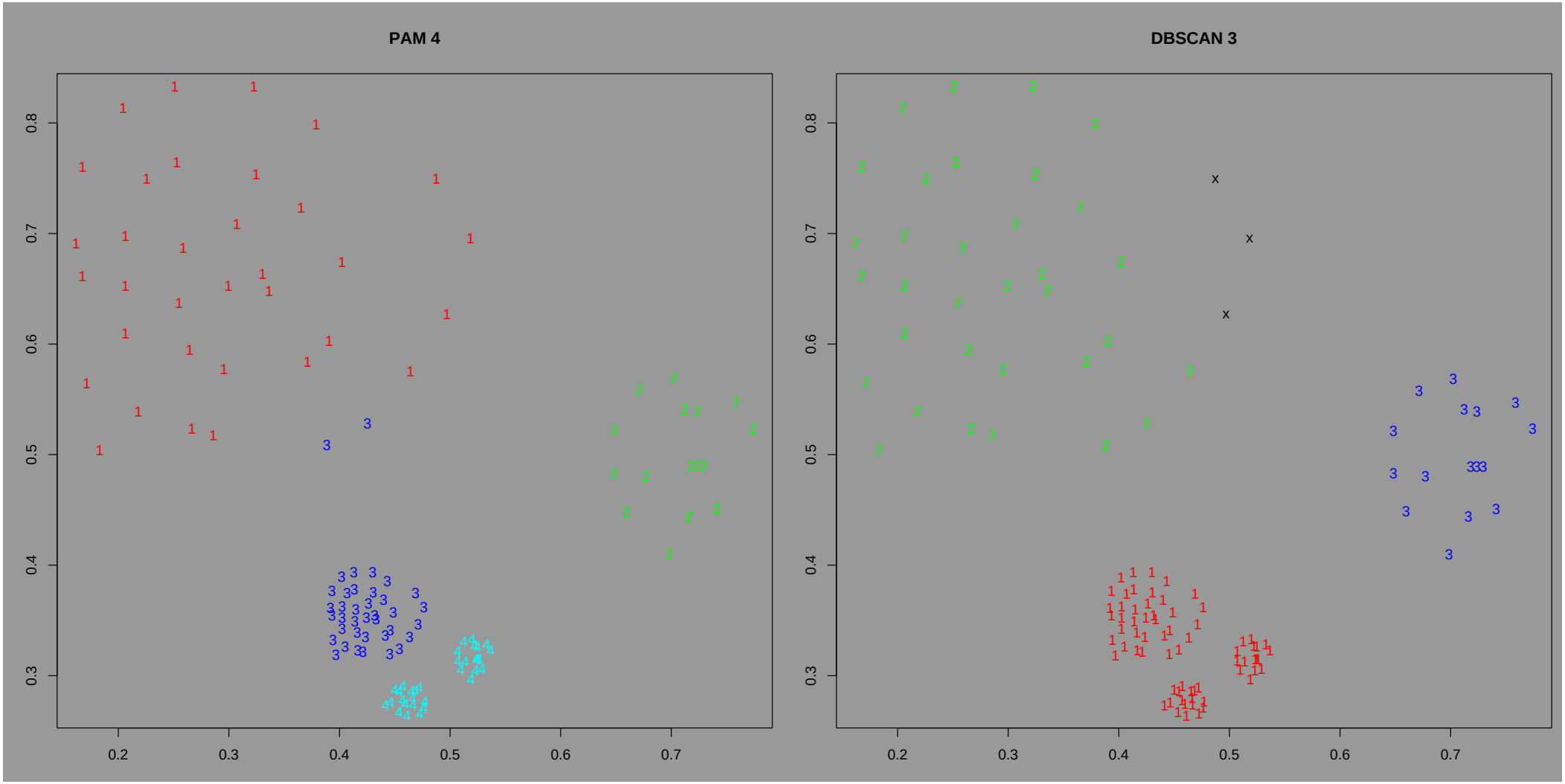
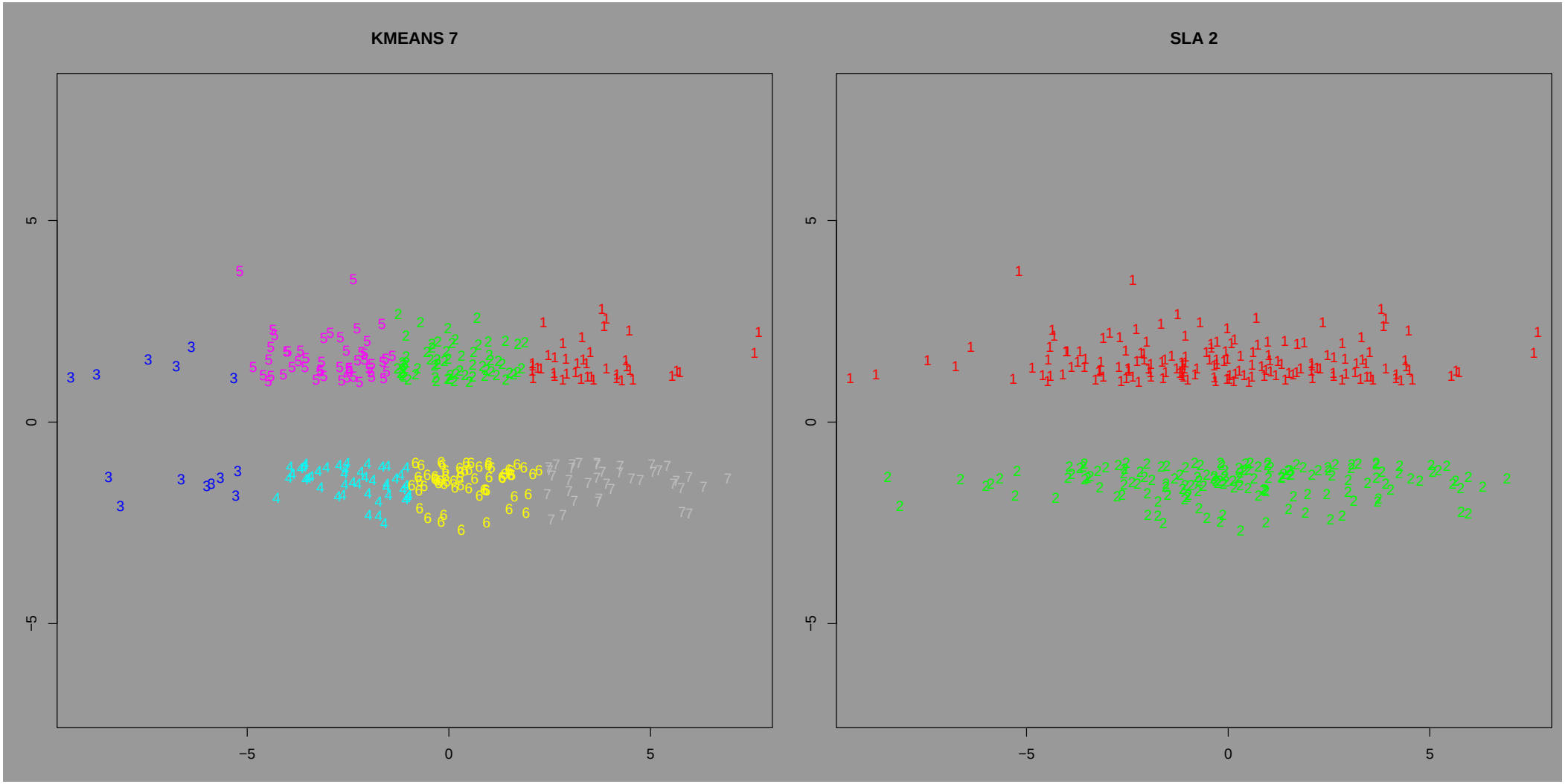


Figure 8: Conflict

Here we have 2 well seperated convex but not spherical clusters. Thus the cluster definitions KMEANS and PAM looking for equally sized rather spherical clusters fail. However, using the non-parametric definitions SLA or DBCAN, truecluster easily identifies the obvious 2-cluster setting.



Conflict continued . . .

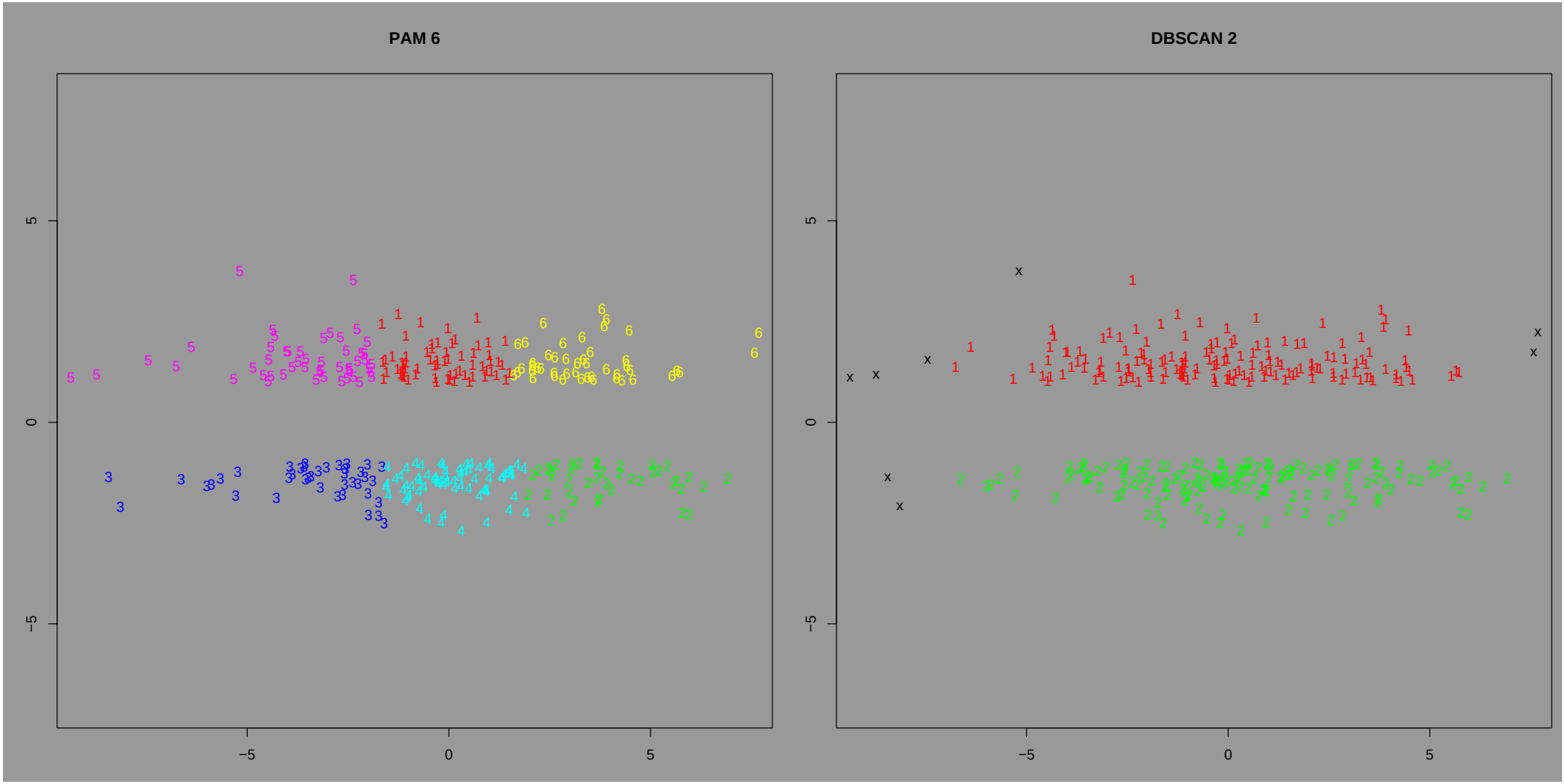
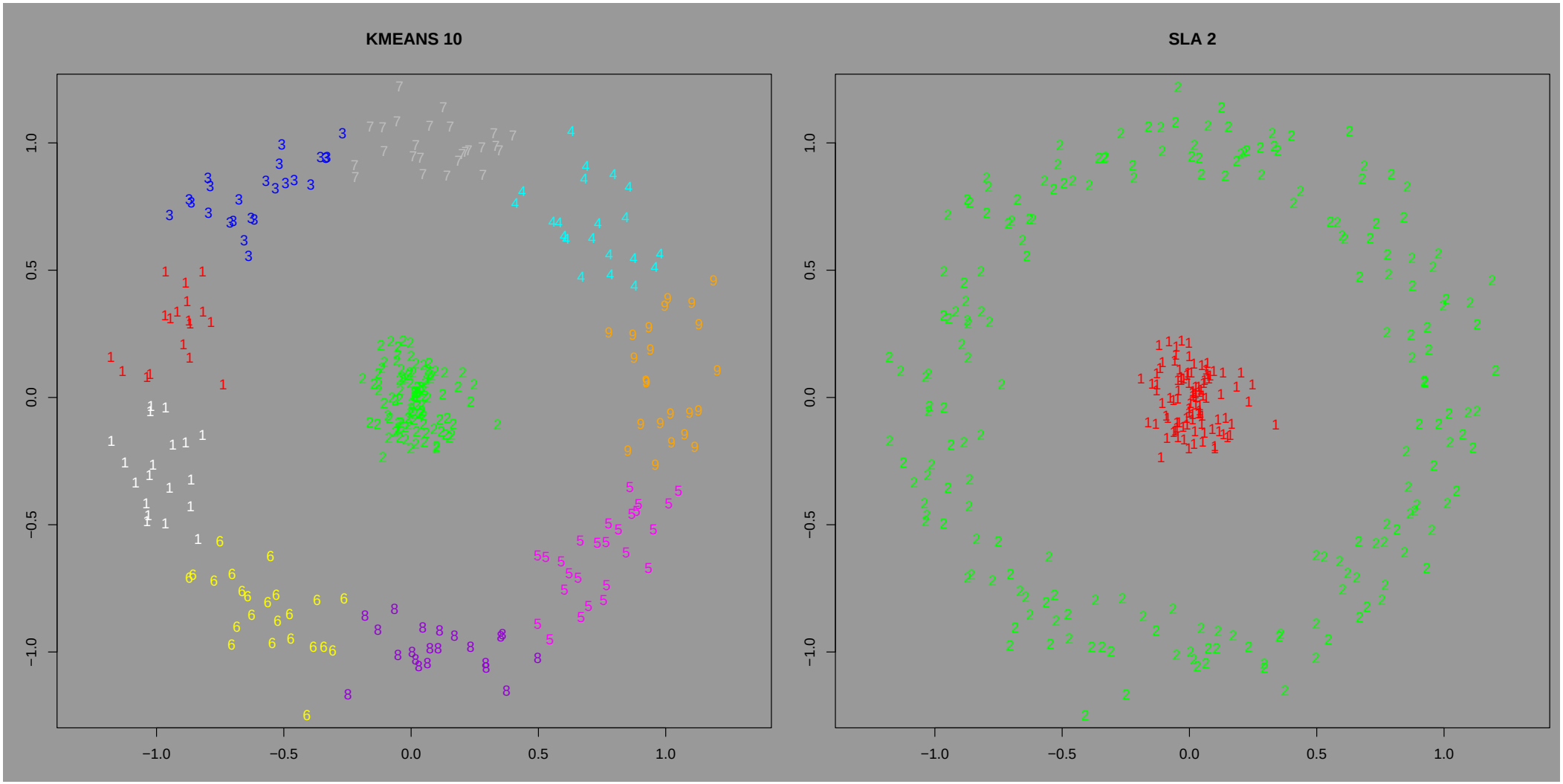


Figure 9: CenterRing

Here we have one cluster surrounded by another. Of course looking for equally sized rather spherical clusters does not make sense and the 'optimal' KMEANS and PAM solutions don't make sense either. However, as these two clusters are well seperated, with both non-parametric definions SLA and DBSCAN truecluster finds the two clusters.



CenterRing continued . . .

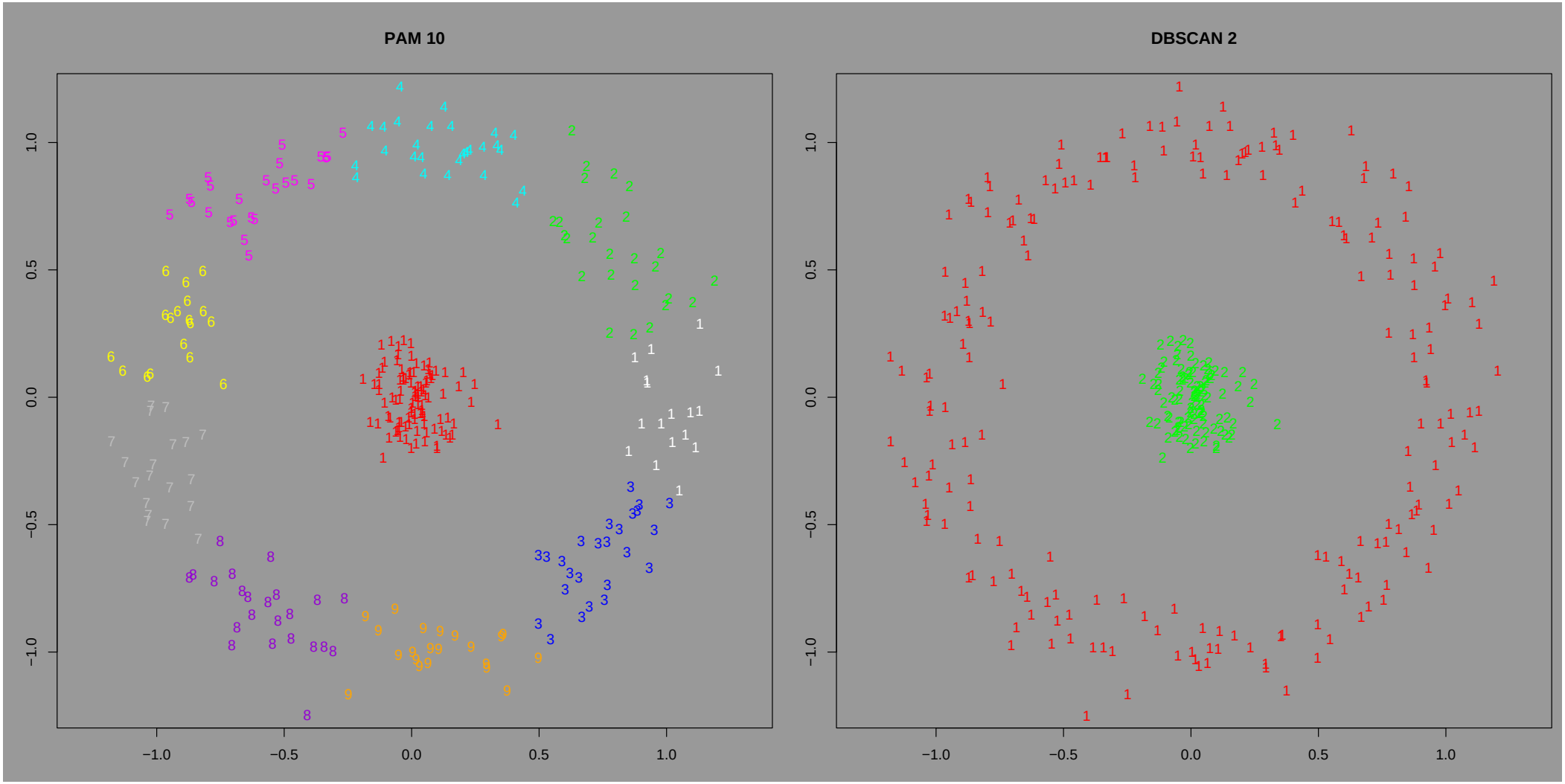
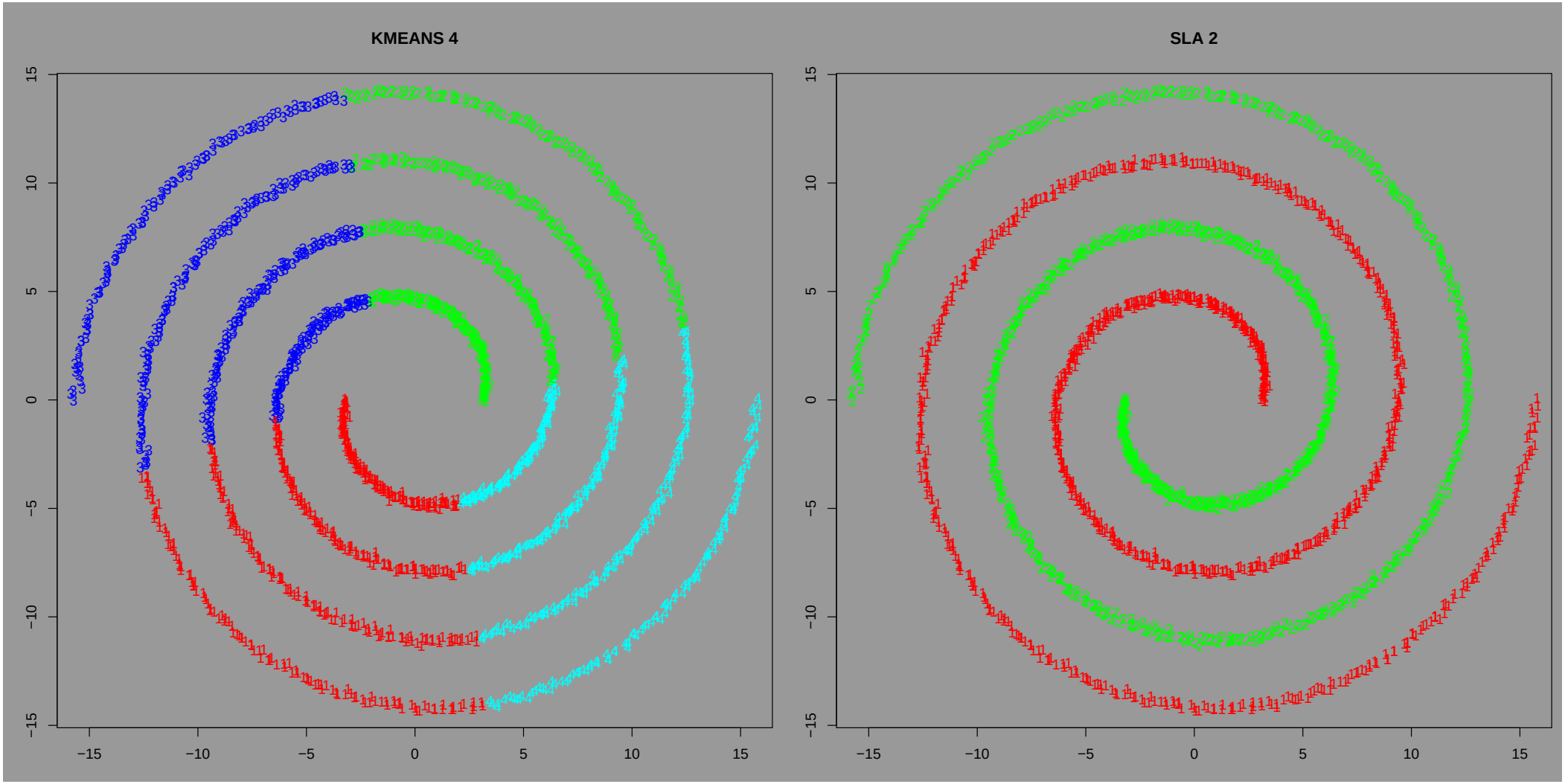


Figure 10: Spiral

The two parametric definitions are obviously inadequate for describing this spiral pattern. Using the non-parametric definitions, truecluster find the two structures unambigously.



Spiral continued . . .