4 Question 4

4.1 Purpose of the function & algorithm's steps

The function `clusterboot` from `fpc` package is an integrated function to assess cluster-wise stability by resampling, which both performs clustering and evaluates the produced clusters. The function enables a number of clustering methods by the interface functions, including `kmeans`, `hclust` and `mclustBIC`.

The algorithm uses the Jaccard coefficient, a similarity measure between sets that measures the ratio of the number of intersection elements over union of the two sets. Specifically, the mean maximal Jaccard coefficient ($\bar{\gamma}_c$) between the original and resampled data sets is computed to reveal the stability of clusters that have been found by the interface clustering method. The basic steps are as follows if we take bootstrap resampling method as an example:

1. Apply clustering method on the data.
2. Draw a bootstrap resample of the same size with replacement from the original data. Obtain clusters on the new data set.
3. For each cluster in the original data, find the most similar cluster that has the maximal Jaccard coefficient. Use 0.5 as the critical value to indicate whether the original cluster is dissolved (< 0.5) or recovered (≥ 0.5).
4. Repeat step 2 and 3 for $i = 1, 2, ...B$ (where B is the number of resampling).

Subsequently, the mean maximal Jaccard coefficient is computed for each cluster to assess the stability. As a rule of thumb, the value higher than 0.75 indicates a valid and stable cluster and value goes above 0.85 suggests a highly stable cluster. The value between 0.6 and 0.75 indicates a certain pattern in the data but not an assured stable cluster. Clusters with the value below 0.6 are considered unstable.

4.2 Simulation and comments

To evaluate how stable a cluster is in the data via the resampling methods that are available in the `clusterboot` algorithm, a data set is simulated and several "true" clusters as well as

"noise" data are constructed. In the simulated six-dimensional data set, the first 4 dimensions that consist of a mixture of 3 Gaussians of different sizes, means and variances are used to generate 3 clusters; and the other 2 dimensions are random values generated from Gaussian distribution $N(0,1)$ and F distribution $F(10,20)$ to simulate outliers. We use the R function `rmvnorm` from the package `mvtnorm` and functions `runif` and `rf` to generate random numbers following a specific distribution. Table 5 shows distributions for 3 clusters and one group of noise data that do not belong to any cluster in the first four dimensions.

Table 5: Data simulation for the first 4 dimensions.

	Data generated from random distribution
Cluster 1	1000 samples following Gaussian distribution with mean vector (0,0,0,0) and covariance matrix of variance 0.2 and zero covariance
Cluster 2	500 samples following Gaussian distribution with mean vector (3,2,3,2) and covariance matrix of diagonals 0.5 and off-diagonals 0.4
Cluster 3	400 samples following Gaussian distribution with mean vector (-2,2,-2,2) and covariance matrix of diagonals 0.5 and off-diagonals -0.4
Noise	20 samples following uniform distribution on [-4,5]×[-2,5]×[-4,5]×[-2,5]

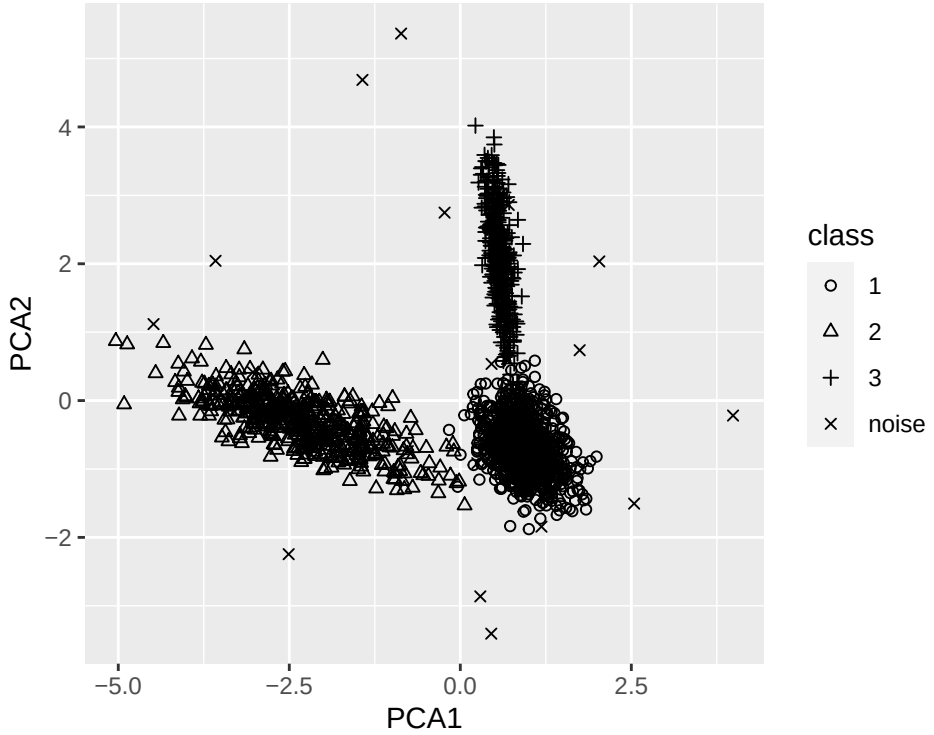

Figure 7: Simulated data represented in a two-dimensional space.

The simulated data of $n = 1920$ observations in total are present in a two-dimensional space as shown in Figure 7. The 3 "true" clusters follow multivariate normal distributions of different sizes and means, among which cluster 1 has no covariance and cluster 2 and 3 have different off-diagonal elements in the covariance matrix. The data set is analysed by three clustering methods, which are k-means, hierarchical and normal mixture model based clustering plus noise. For a given cluster analysis method, `B=100` number of resampling is run for each scheme and the average maximum Jaccard coefficient ($\bar{\gamma}_c$) is recorded. The methods used for resampling are bootstrap ("boot"), subsetting a size of $n/2 = 960$ subsamples ("subset"), replacing 0.05 of points by random noise from uniform [-2,2] ("noise") and adding 0.1-quantile of the 1-dimensional distances between neighboring points to every single point ("jitter").

The 3-means and 4-means clustering methods are carried out on the unscaled data via the interface function `clustermethod=kmeansCBI` and $\bar{\gamma}_c$ of 100 resamples under 4 resampling methods are shown in Table 6 and Table 7. The 3-means clustering generates "highly stable" clusters as all the mean Jaccard similarity values are no less than 0.85. The recovery of each cluster is around 78~89 and close to 100, which is also an indicator of stable clusters and detected by all resampling methods. The size of cluster 1, 2 and 3 is found as 1039, 486 and 395, which are close to the true cluster size 1000, 500 and 400. The 4-means clustering tends to split cluster 1 into two and generate less stable clusters with $\bar{\gamma}_c$ ranging from 0.680 to 0.780. The low stability is also verified by the low number of times that cluster 1 has been recovered (42~57) and the high number of times that cluster 1 has been dissolved (43~58). Overall, k-means clustering is a fast and simple method to apply but the choice of k value is critical to generate stable clusters. Additionally, the 3 clusters are distinct from each other by the mean of Gaussian distribution in dimension 1 to 4, which enhances k-means' ability to recover the true clusters.

Table 6: The mean Jaccard coefficients resulted from different resampling methods for 3-means clustering.

label	boot.mean	subset.mean	noise.mean	jitter.mean
1	0.933	0.965	0.962	0.941
2	0.899	0.949	0.954	0.911
3	0.850	0.921	0.911	0.863

Table 7: The mean Jaccard coefficients resulted from different resampling methods for 4-means clustering.

label	boot.mean	subset.mean	noise.mean	jitter.mean
1-1	0.780	0.765	0.716	0.729
1-2	0.752	0.734	0.680	0.695
2	0.883	0.871	0.870	0.822
3	0.891	0.888	0.861	0.873

Table 8 shows the mean Jaccard similarities for hierarchical clustering with average linkage via the interface function `hclustCBI`. The partition is obtained by cutting the tree that results in 3 clusters. The average linkage partition seems to find cluster 1 successfully as $\bar{\gamma}_c$ values are above 0.975 for all resampling schemes. However, the mean Jaccard similarity vary greatly from using different resampling methods for cluster 2 and 3, indicating the non-robustness of the clustering. The $\bar{\gamma}_c$ values are rather low for cluster 2 among all resampling schemes except jittering, which is said to provide bad indication for average linkage partition (Hennig, 2007). Closer inspection exhibits that the hierarchical clustering partitions as many as 1909 observations into cluster 1 and thus generates a poor solution.

Table 8: The mean Jaccard coefficients resulted from different resampling methods for hierarchical clustering (average linkage).

label	boot.mean	subset.mean	noise.mean	jitter.mean
1	0.979	0.975	0.997	0.999
2	0.520	0.447	0.634	0.758
3	0.747	0.708	0.877	1.000

Another clustering is obtained by a normal mixture model based clustering with a noise component and the $\bar{\gamma}_c$ values are shown in Table 9. We use the interface function `noisemclustCBI` and specify `G=NULL` to evaluate 1-9 number of clusters and select the optimal value by the BIC. We also specify `nnk=5,noisemethod=TRUE` to apply 5 nearest neighbor denoising. Additionally, `modelNames=c("VVE","VVV")` is specified with prior knowledge to add size/shape/orientation constraints to the covariance matrix of Gaussian mixture models (Scrucca et al., 2016). Based on the BIC, the (VVE,2) model is selected such that two clusters are described by Gaussian components with varying volume and shape but same ori-

entation aligned with the coordinate axes. The last cluster is detected as noise but unstable under jittering resampling. Further investigation shows that out of 27 detected noise points, only 7 belong to the true noise cluster. The true cluster 2 is highly stable under all resampling schemes, while points from the true cluster 1 and 3 are classified as one cluster when choosing VVE model and thus the clustering is considered stable under bootstrap, subsetting and replacing points by noise but unstable under jittering.

Table 9: The mean Jaccard coefficients resulted from different resampling methods for normal mixture model based clustering plus noise.

label	boot.mean	subset.mean	noise.mean	jitter.mean
1	0.996	0.993	0.994	0.511
2	0.999	0.999	0.999	0.994
3 (detected as noise)	0.825	0.756	0.775	0.242

To summarise, different clusering on the same data can result in clusters of different level of stability. Our results show that a high value of $\bar{\gamma}_c$ indicates a stable cluster but not always meaningful. The average linkage hierarchical, for example, partitions 1909 observations into one cluster and gets a rather high $\bar{\gamma}_c$ value. The k-means clustering that is fast to apply behaves well when the optimal k value is chosen and when the true clusters are distinct from each other in terms of the cluster mean. Gaussian mixture models based clustering provides the benefit of identifying the optimal number of clusters given certain constraints to the covariance matrix. It is also noticeable that among the 4 resampling methods that have been used in this analysis, jittering usually gives different indication of stability compared to the other 3 resampling schemes.