NSERC URSA Summer 2019 Summary

| Shouyang Zhou  University of Alberta  Edmonton, Canada  shouyang@ualberta.ca |
| --- |

# Introduction

This report summarizes my experiences under the NSERC undergraduate student research awards program of summer 2019. My term consists of two projects with PhD Student, Georg Steinbuss of Karlsruhe Institute of Technology and Professor Jorge Sander of University of Alberta.

The two projects I’ve worked on deal with outlier detection methods and benchmarking datasets used for outlier detection methods. The first project looked at using kernel density approaches for outlier detection. The second project looked at examining commonly used comparison datasets used to evaluate outlier detection methods.

This report will be organized into two primary sections, one per project. Each project section will describe some preliminaries to the project, approaches taken, and findings accordingly.

# Project One – Kernel Density Approaches To Outlier Detection

**Preliminaries**

Kernel density estimation (KDE) aims to estimate the probability density of some dataset non-parametrically. If KDE can provide a reliable estimate of the originating distribution, then it can prove useful as a contrast for determining the outlierness of an observation in question.

I supported Georg Steinbuss in investigating the use of kernel density estimation approaches to devise an outlier detection method. I conducted initial experiments of various R and Python KDE methods. My experiments focused on the accuracy of the select KDE methods in 1-D / 2-D spaces where there were intuitive benchmarking datasets.

We selected libraries of interest from a survey article “Density Estimation in R” (Deng & Wickham, 2011). They examine various KDE implementations in R and provide runtimes and accuracy measures, accuracy by mean absolute deviation from a known distribution. One package titled “np” was promising as it fit the criteria of having few parameters and being able to support an arbitrary number of dimensions. Many packages surveyed by Deng & Wickham where either univariate or severely limited in dimension. The next best package supported up-to six dimensions. Over the course of the project, Georg found three other approaches of interest (fastkde, mclust, knn-kde).

**Approaches Examined**

* **Np package**: The np package provides various non-parametric procedures for an assortment of purposes. Its main audience is econometricians. This library was chosen because its KDE functions was well fleshed out. It supports an arbitrary number of dimensions, various KDE approaches such as cross validation bandwidth selection, multiple kernel types, and adaptive kernel functionality.
* **Fastkde**: This package claims to be a self-optimal package that is completely parameter free and a magnitude faster than other KDE approaches. The authors call their approach a “self-consistent KDE in arbitrary dimensions”. Given the promising claims of the package, we decided to investigate this package further.
* **Mclust**: Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation is not strictly a KDE density method. The motivation to compare Mclust was based on the view of density estimation as a spectrum between the use of some singular density (distribution fitting) versus some density for every observation (KDE). Between these, we are curious if fitting a moderate to high number of mixtures can result in an improvement over KDE.
* **Knn-Kde:** KNN-KDE methods were implemented following an investigation of np’s adaptive kernel functionality whose details are skim. Many of the KDE methods use a static kernel making the choice of the kernel and its parameterization key, in contrast KNN-KDE methods use K-nearest neighbors to adapt to the region being estimated potentially improving the estimate. This also included solely KNN based approaches.

In addition to these, the base R::density KDE methods were examined as a grounding point for this project.

The methods selected recommend either some rule of thumb or a cross validation method for parameter selection. Where offered (np, fastkde, mclust), I use the cross-validation scheme provided. For KNN approaches, we use a rule of thumb method described [here](http://faculty.washington.edu/yenchic/18W_425/Lec7_knn_basis.pdf), k = c · n ^ (4/5) where c being some constant, we choose 1.

**Experiments**

The packages were examined for their accuracy in relatively low-density regions of benchmark distributions. The two primary measures were rank correlation and differences between the actual density versus the estimate at various points of the low-density regions. We also consider the performance across sample size, the choice of tuning parameters and the stability of estimates.

Benchmarking densities were selected from the R package “benchden” which implements 28 univariate distributions meant for testing density estimation schemes. I focus upon three benchmarking distributions for discussion, normal, bimodal, smooth comb, as they represent areas where KDE methods are most suitable. The three distributions are smooth continuous distributions that vary in complexity. In general, most other distributions were like these three or they feature sharp areas unsuited for KDE methods. See the appendix on the benchden densities.

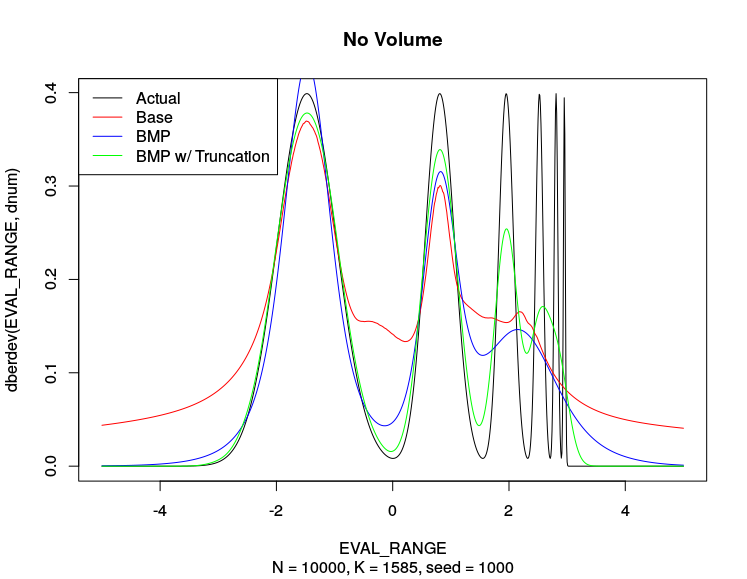
**Results**

This section will first summarize the overall performance of the methods surveyed, and comment upon explorations into multivariate tests using the best candidate method. Secondly, it will discuss some interesting features about the methods outside the main performance measures of rank correlation and differences between estimate and actual density.

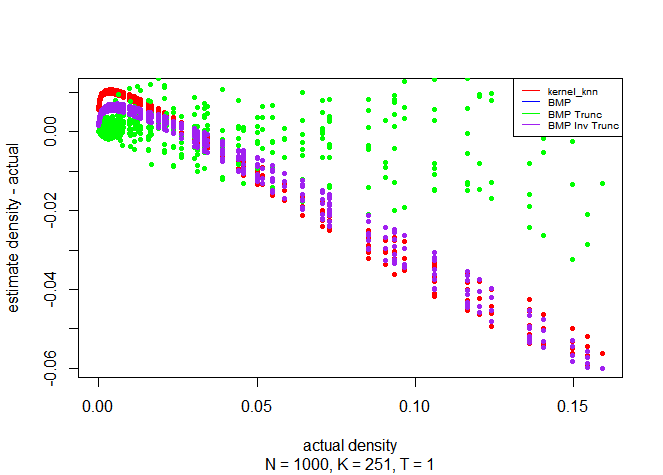
Please see the comparison plots of methods (excluding fastkde) in the appendix. The plots compare either rank correlation or differences squared between the estimate and actual density at a grid of low-density sample points. The family of methods at labelled in brackets. The benchmarking distributions range from relatively simple in complexity on the left to relatively complex distributions on the right.

In general, most methods can estimate the overall shape of the distribution in relatively simple cases (normal, bimodal) but struggle with complex distributions as with the smooth comb distribution. Increasing the sample size does not significantly improve performance but results in greater stability of the estimates.

Overall, the adaptive KNN-KDE method BMP with truncation proved to strike a balance between strong rank correlation and relatively low squared differences to the actual density in these univariate experiments. See [here](https://www.sciencedirect.com/science/article/pii/0047259X9290095W) details on the method.

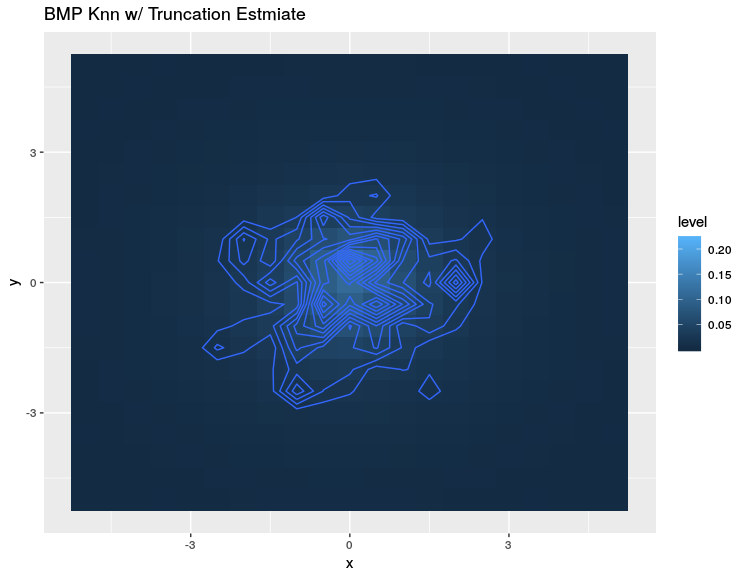


1 KNN-KDE Estimates on Smooth Comb Distribution

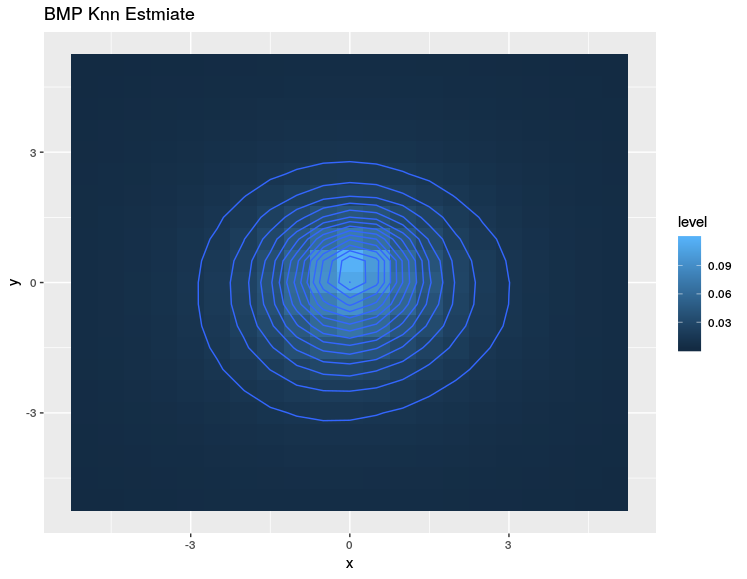


2 Error of KNN-KDE family of methods in separate experiment. BMP Truncation (Green) shows closer estimates in moderate density regions (pdf of 0.05-0.15).

Extending this method, I explored the performance of BMP with truncation in two dimensions on a standard normal multivariate distribution. Unfortunately, the estimates produced via BMP w/ Truncation are poor. At n = 100, the results of truncation leave a jagged and scattered estimate in contrast, as an example, to that of pure KNN density estimate. When n is increased, the performance of BMP truncation appears to be on par with that of regular KNN density estimates. The lack of intuitive benchmarking distributions here was a challenge for further experiments.



3 BMP with Truncation n = 100

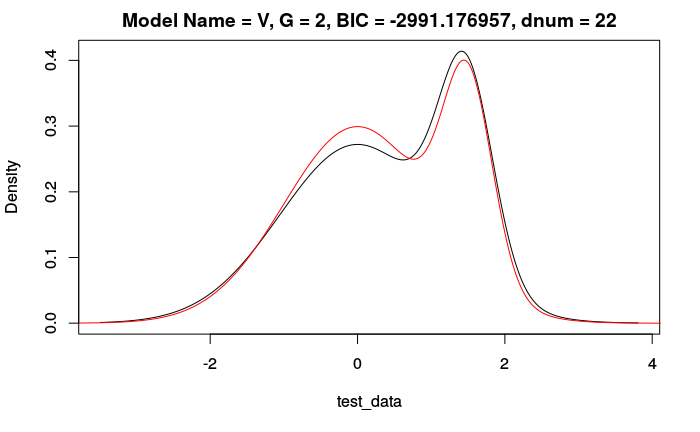


4 KNN Estimate n = 100

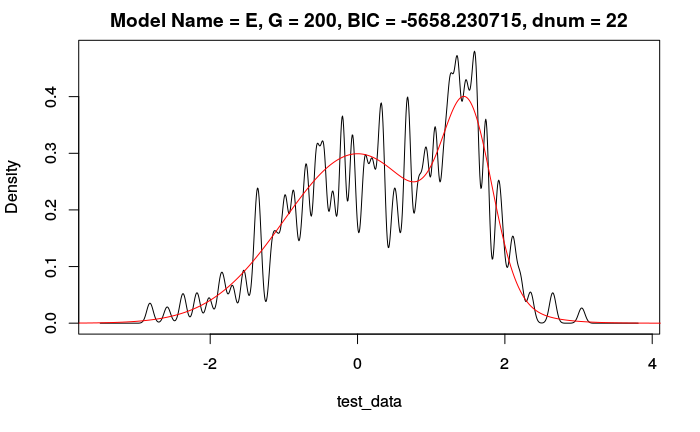
While the BMP with truncation KNN KDE method was effective in one dimension, scaling into higher dimensions demonstrates some initial challenges.

There were two other findings of interest, one with the use of mclust, gaussian mixtures, as a pseudo kernel density estimate, and another with the performance of fastkde.

In the summary plots, mclust performed strongly in the normal and bimodal scenarios however significantly poorly in the smooth comb distribution. Mclust is effective when the distribution is relatively simple however it faces difficulties when required to include more mixtures considering complex distributions. When mclust can choose between 1-n number of mixtures, it favors lower mixture counts with slight increases accommodating the complexity of the distribution. When forced to use a high number of mixtures, as to represent a sort of kernel density estimate, mclust creates very unstable jagged estimates.

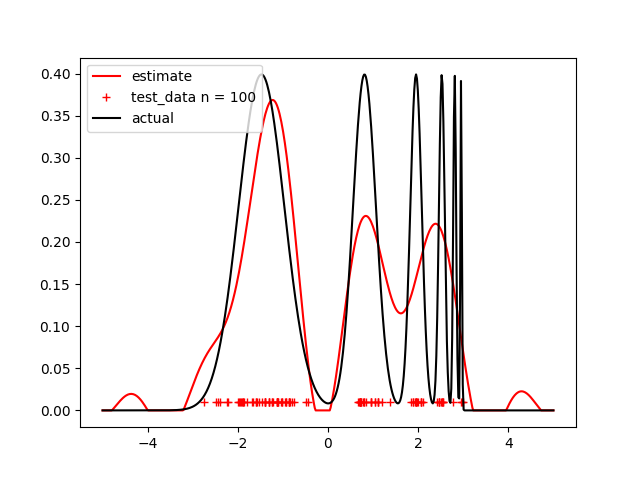


5 Estimate when choosing between 1- n =1000 mixtures, choice = 2.

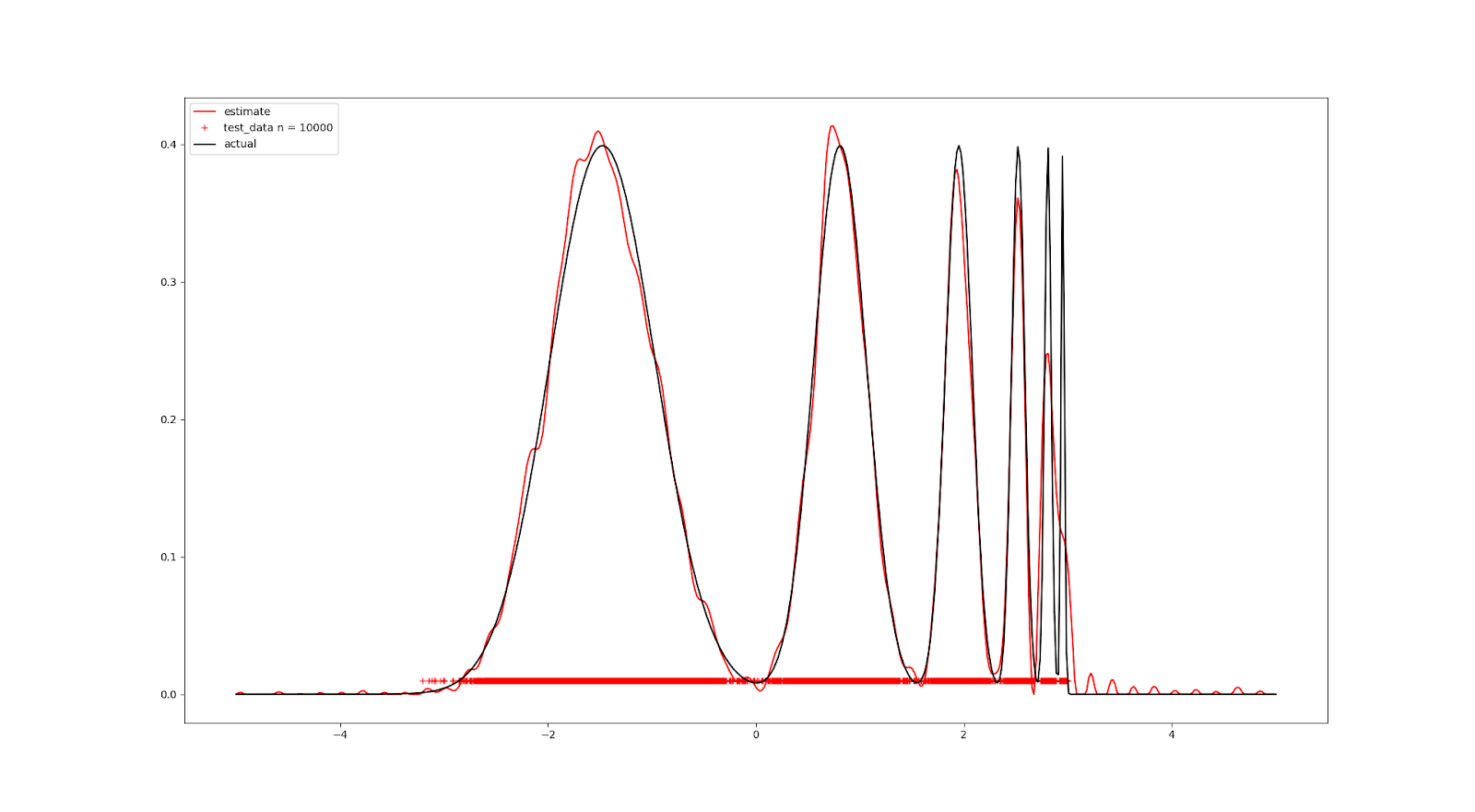


6 Estimate when number of mixtures fixed to 200.

With fastkde, I found that the estimates produced contained odd “bumps’ in the tails of distributions that made it unsuitable for outlier detection purposes. This effect is especially notable when n is small. When n is large, fastkde produced accurate estimates of the overall distribution but these tail-end results remained present albeit diminished.



7 fastkde estimate of smooth comb distribution n = 100



8 fastkde estimate of smooth comb distribution n = 10000

# Project Two – Exploratory Data Analysis of Outlier Detection Datasets

**Preliminaries**

This project examined commonly used benchmarking datasets as per the paper “On the Evaluation of Unsupervised Outlier Detection: Measures, Datasets and an Empirical Study”. This project was motivated to better characterize and assess the suitability of the benchmarking datasets. Because there is no standard for benchmarking outlier detection methods, it is difficult to compare and characterize differing approaches to outlier detection.

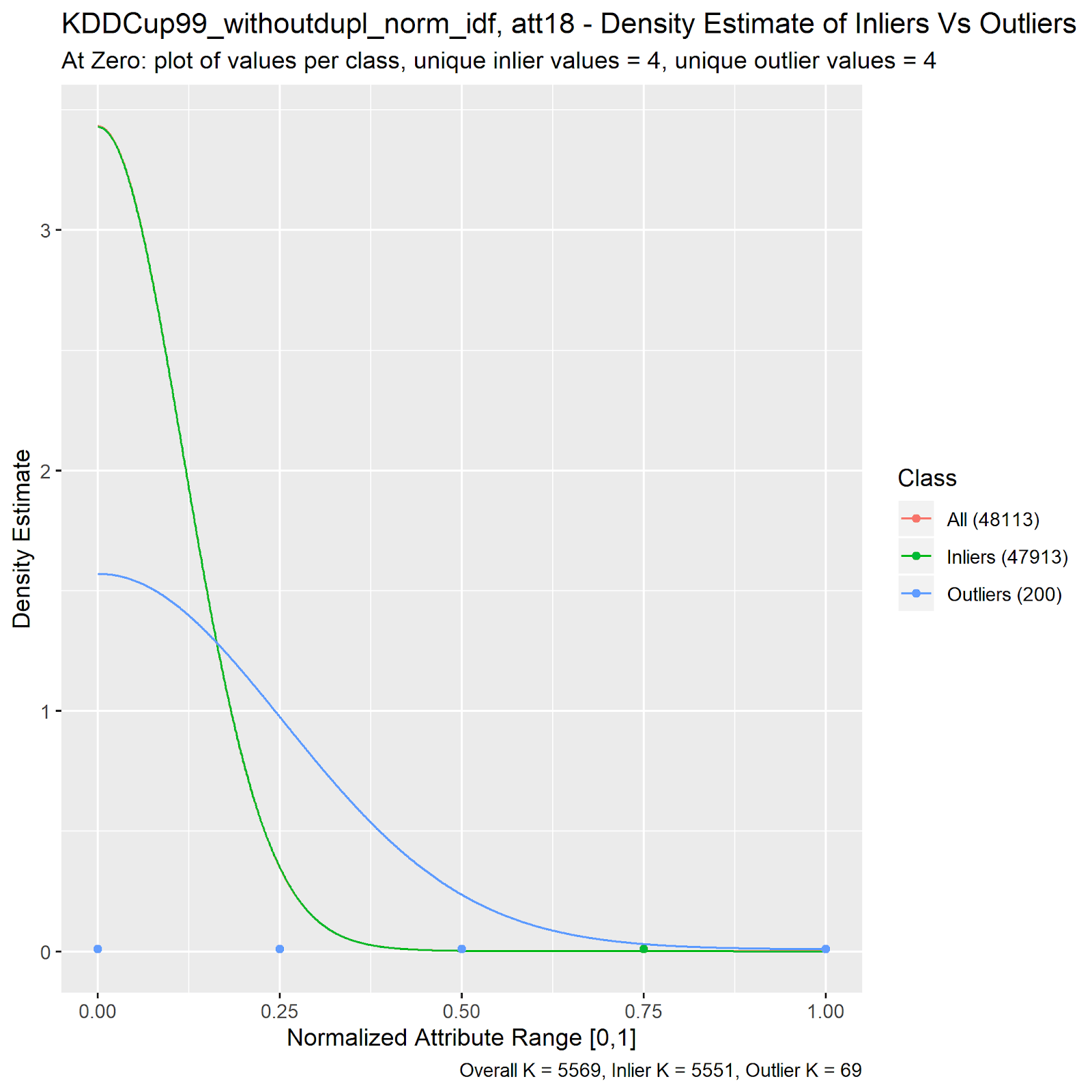
The datasets examined represent either datasets used in outlier detection literature or semantically meaningful reference datasets used for classification or some other benchmarking purpose. The notion of an outlier in these datasets are generated, usually, by denoting minority classes as outliers and sampling.

The datasets presented had many variants which accommodated differing proportions of outliers, attribute scaling and normalization, and the handling of categorical values and duplicate observations. Where possible, I examined the datasets that were normalized, had no duplicates, used idf encoding of categorical variables, and contained either five or ten percent outliers.

**Approaches Examined**

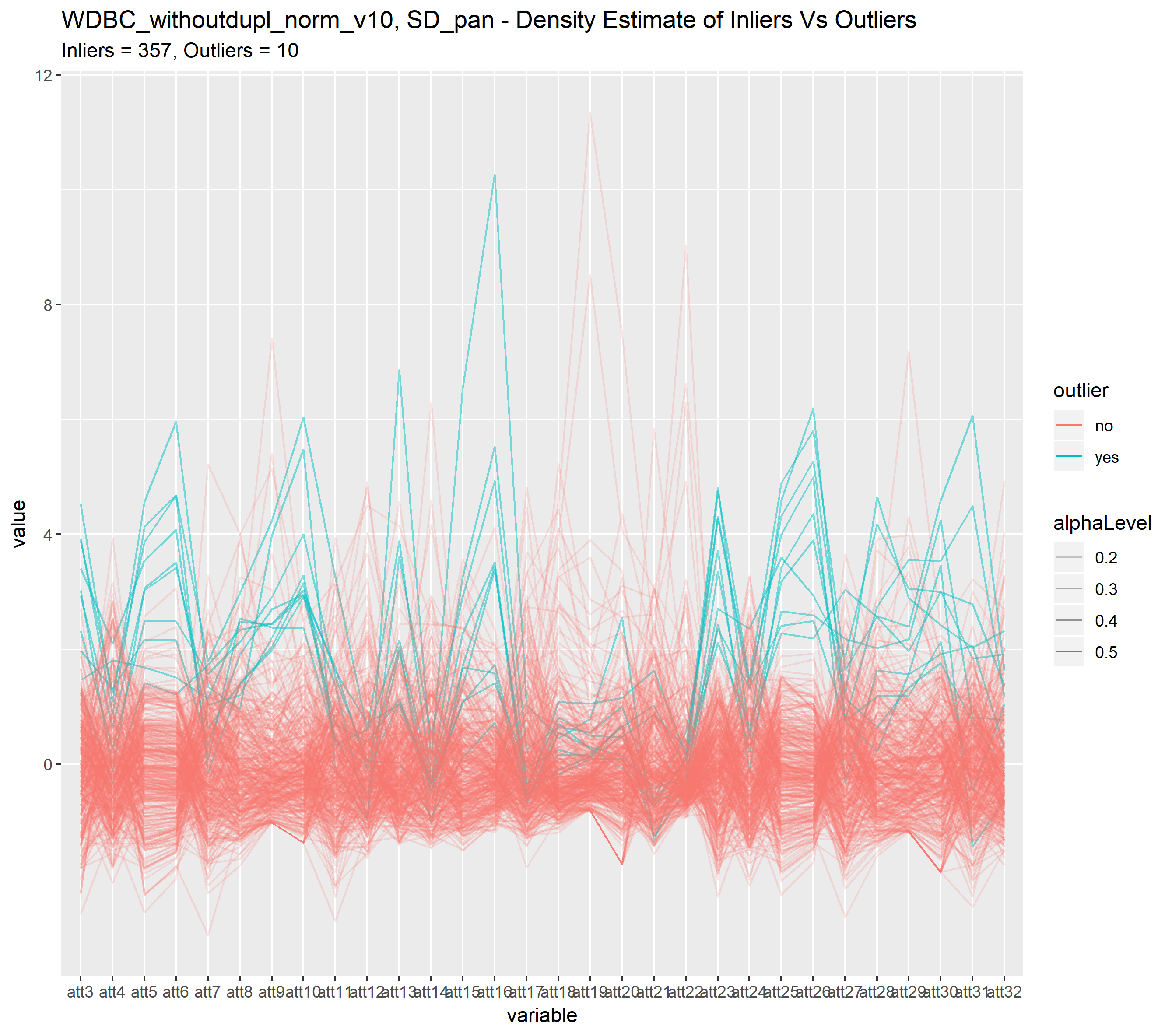
* **Kernel Density Estimation:** Extending from the first project, KDE methods were used to first summarize the datasets. Georg mentioned that the resolution of the datasets was an area of interest. In addition, KDE could be used to compare the separability of outliers to inliers. As BMP with truncation proved to be an effective one-dimensional estimation method, this method was used per attribute per dataset.
* **Parallel Plots:** Parallel plots were used to visualize the datasets and the separability of inliers to outliers. These act as a complementary to note the separability between inliers and outliers of the datasets. The KDE estimates give an idea of individual attributes per dataset while the parallel plots give context for the entire dataset.
* **K-Means & HDBSCAN:** Two clustering methods were applied to the datasets. As many outlier detection methods surveyed has some basis in a nearest neighbors or density-based method, k-means and HDBSCAN clustering could provide some insight on how various methods approach each dataset. Clustering was performed by varying the number of clusters such that 2-15 clusters would form to examine the decomposition of clusters as k increases. For HDBSCAN, this was done by traversing cut points such that the first cut to generate k clusters would be used. HDBSCAN was ran using default settings.
* **Silhouette Scores:** To complement the clustering results, silhouette scores which measure the consistency of clusters were generated. The contrast of the average cluster scores for outliers versus inliers may be of interest.

**Results**



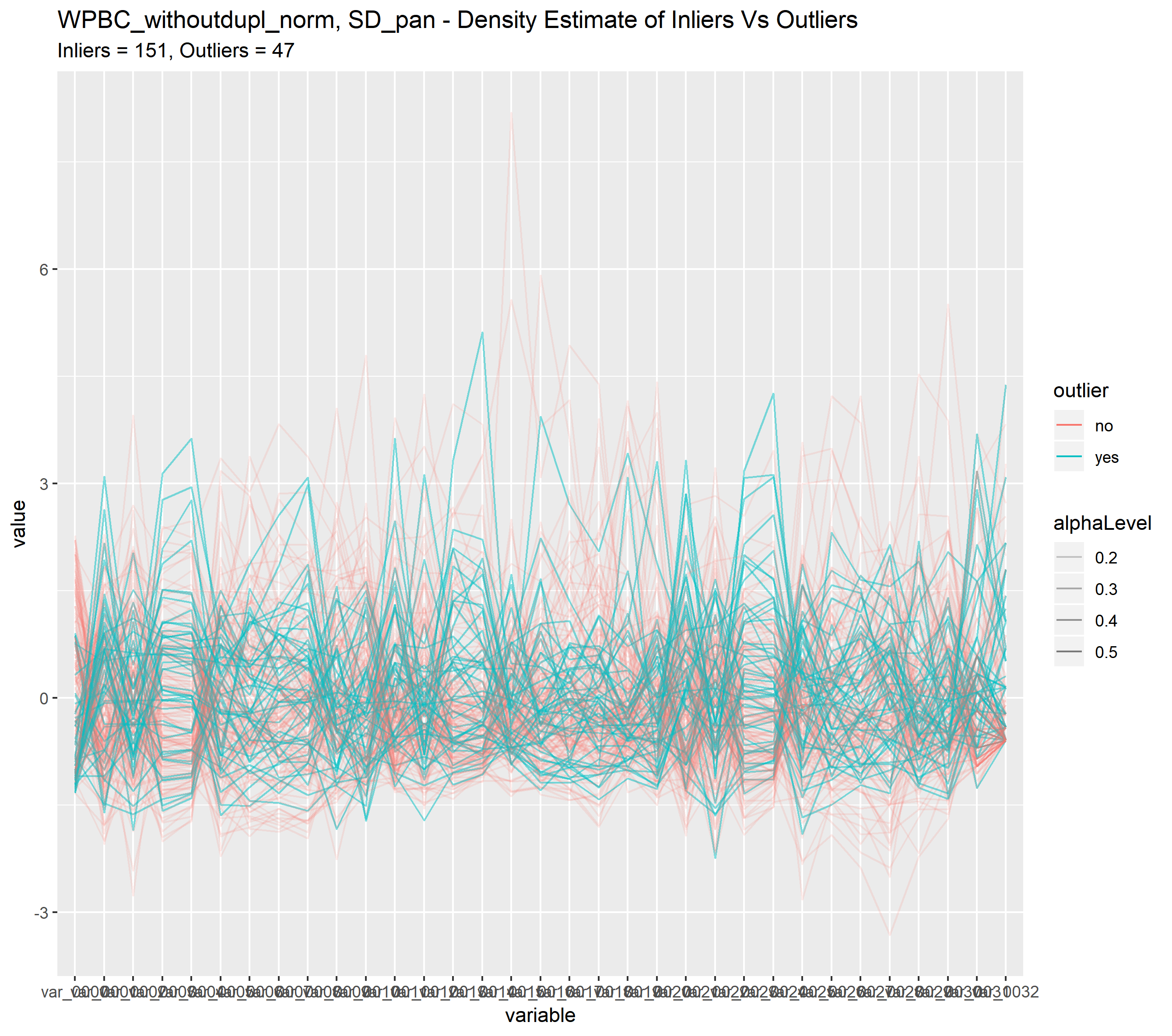
9 Example Kernel Density Estimate Plot

KDE found that often there was no clear separation of outliers in individual attributes alone but the complementary visualization of unique values per inlier and outlier groups shows that the normalized datasets contained various low resolution, binary, or otherwise “stepped” like attributes. Datasets in the semantic category had a higher prevalence of these attributes than the ones in the literature category. A dataset that included one binary attribute included multiple. In some of these low-resolution attributes, KDE hinted towards that outlier labelling placed higher importance in some attributes than others.



10 Parallel Plot of The WDBC Dataset – Outliers in Teal

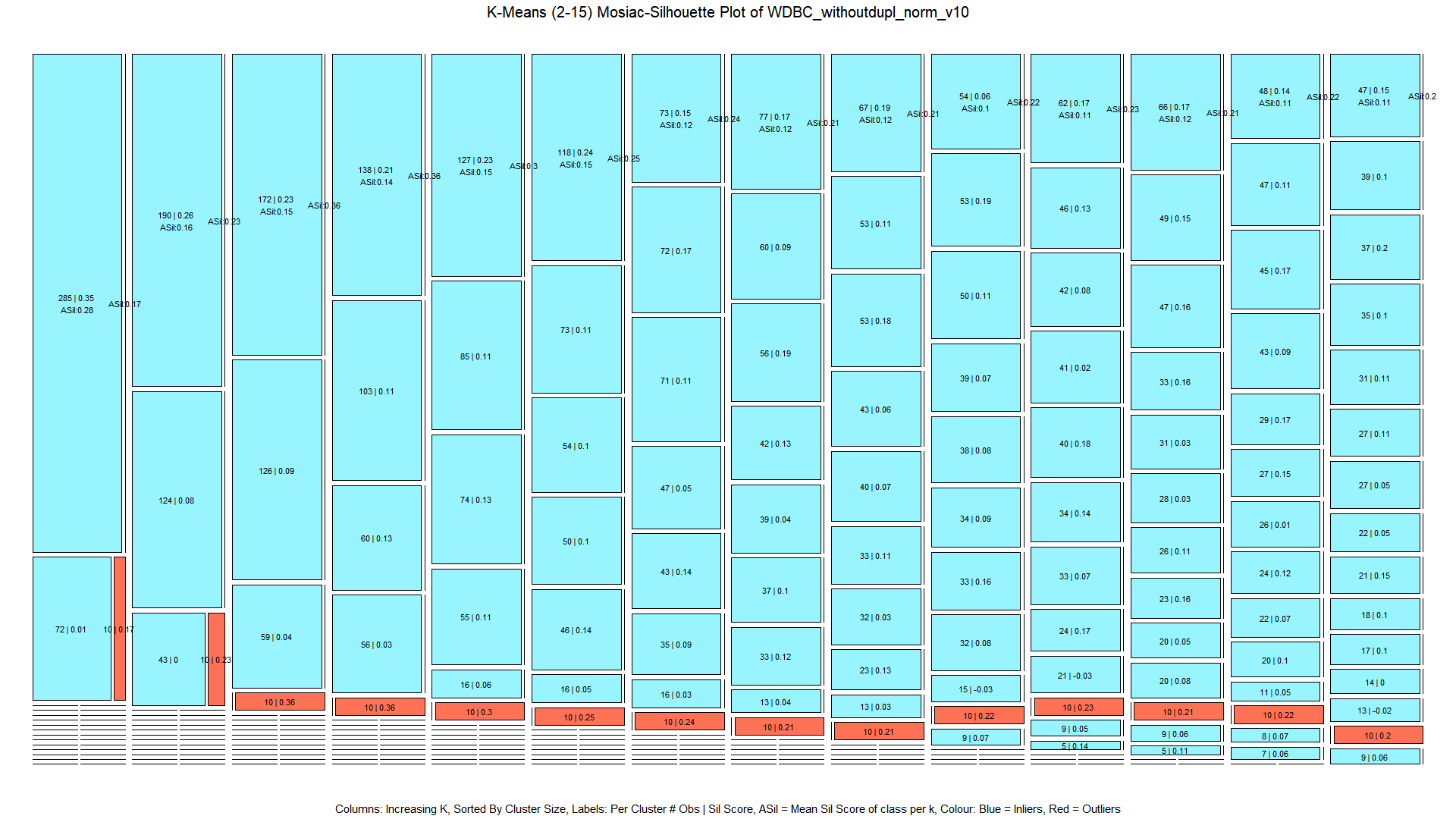
Parallel plots examined the outlier labelling further. If outliers are labelled adequately then we should expect outliers to be at the edges of labeled inliers or to deviate structurally to some trend with inliers. Most datasets found generally agreeable labelling however as demonstrated by the WDBC dataset, there are often equally interesting labelled inliers. There is a range of attributes where some inliers have extremely high values in contrast to the rest of the dataset.



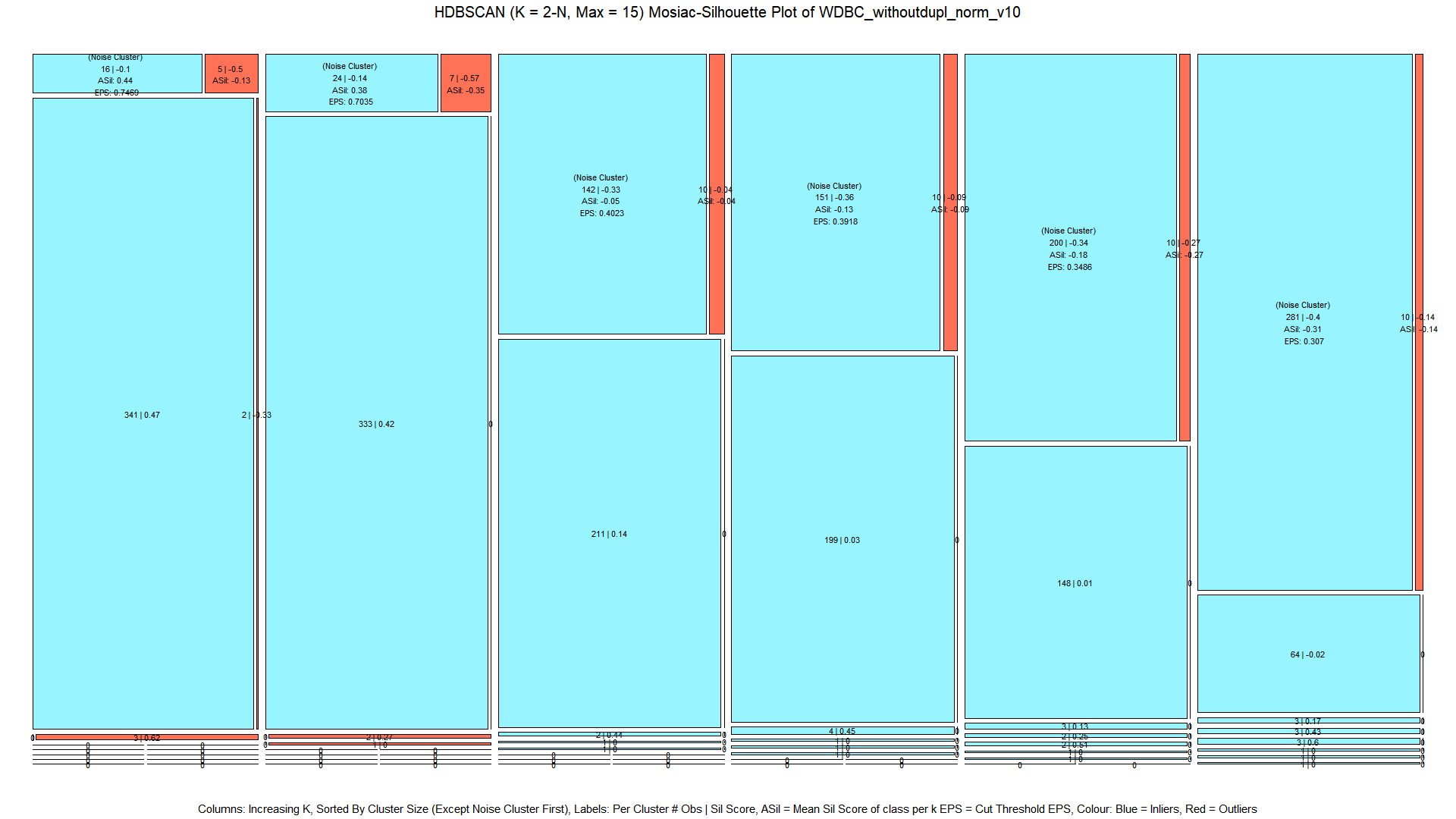
11 Parallel Plot of The WPBC Dataset

Parallel Plots found that the WPBC literature dataset may not be appropriate for benchmarking regular outlier detection schemes. This dataset had the best ROC\_AUC score of approximately 0.58 of the methods surveyed. The source of this dataset was a paper noting a novel data processing method.

It may be interesting to use parallel plots to compare contextualize the outlier scoring of a given method or family of methods. For example, some coloring scheme may be helpful in extending the notions of difficulty and diversity presented in the survey paper by visualizing where within a dataset outlier detection methods have difficulty or disagreement.



12 K-Means Clustering Of WDBC



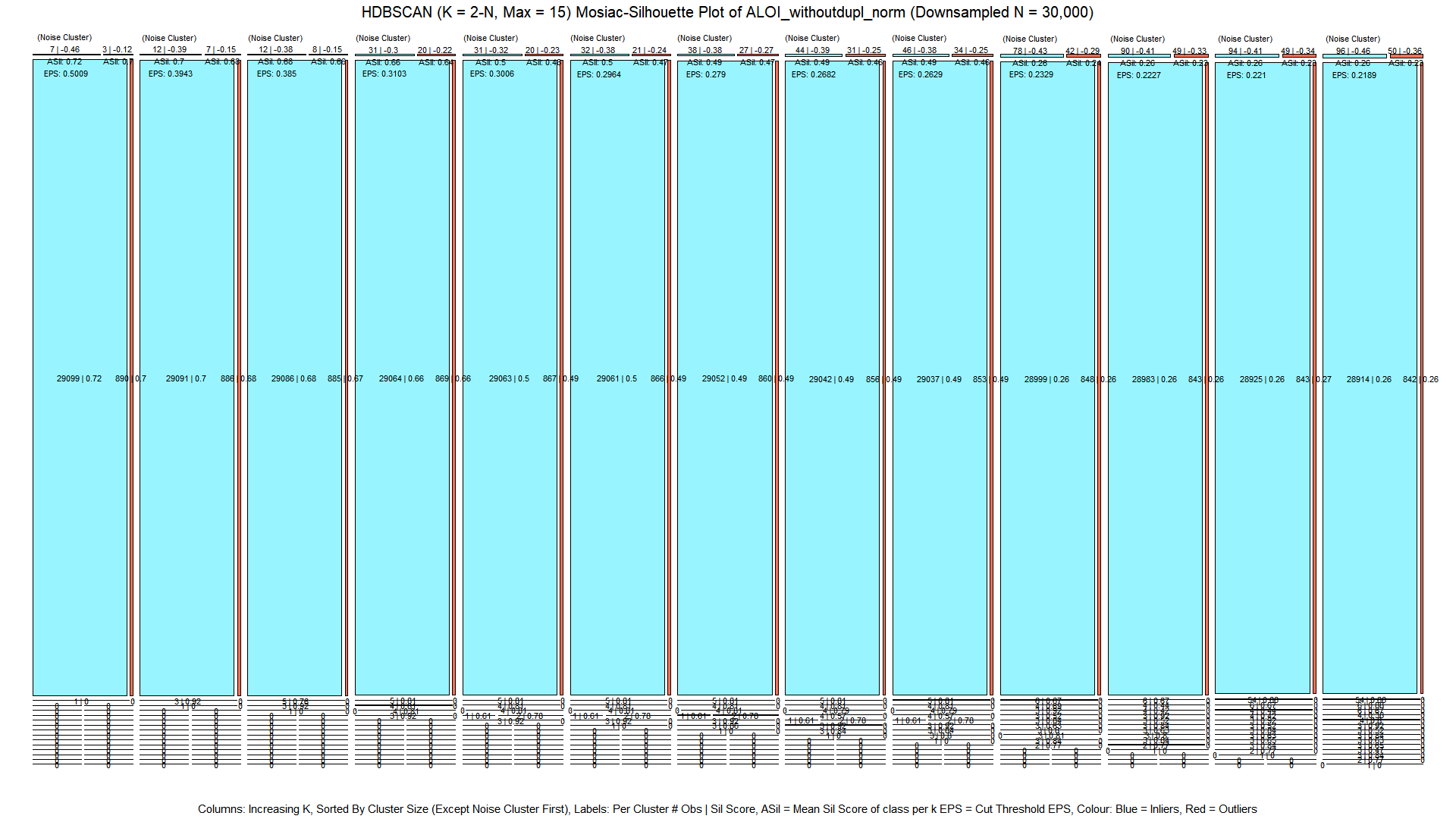
13 HDBSCAN Clustering Of WDBC

Lastly, clustering methods and silhouette scores are combined into mosaic plots that visualize how clusters form of inliers (teal) and outliers (red) over an increasing number of clusters. Going from left to right, clusters are arranged in increasing k, ordered by the size of the clusters. The silhouette score is presented for the cluster’s inliers and outliers separately. The global average silhouette score for inliers and outliers are presented below the first cluster. For HDBSCAN, the first cluster is the grouping of all noise points which includes also the eps cut value used to obtain the clustering.

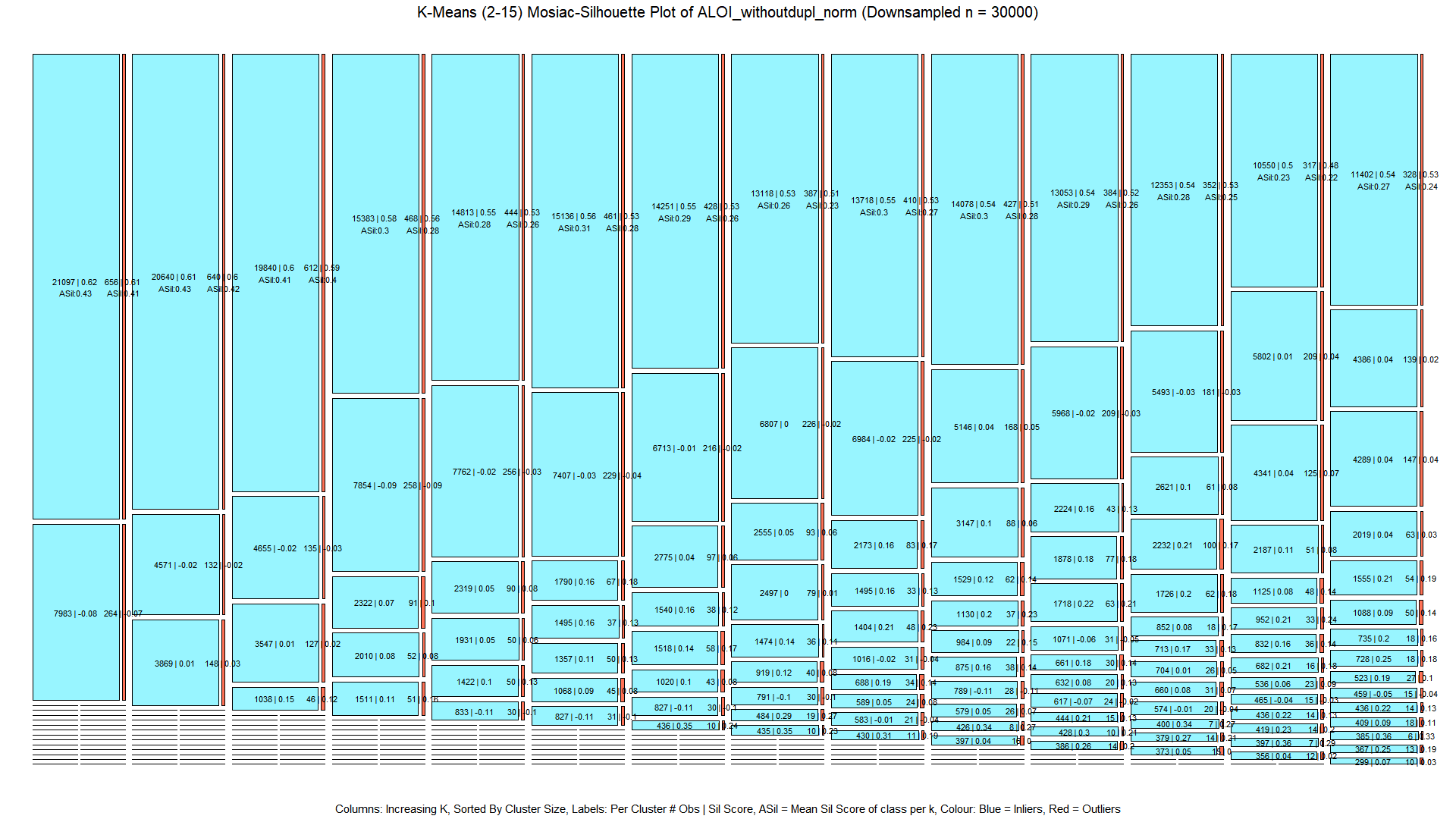
This visualization helps verify the difficulty scores of the datasets as seen in the survey paper. The ease of outlier detection can be seen in three forms. First, the labelled outliers form a minority cluster as k is increased. Second, the contrast of silhouette scores within a cluster. Third, for HDBSCAN, the concentration and proportion of outliers in the noise cluster.

Easier datasets such as WDBC, WBC, Parkinson, and Lymphography are characterized by the concentration and formation of outlier clusters in K-means / HDBSCAN and the concentration of outliers in the noise cluster at the highest eps value of HDBSCAN.

The hardest datasets such as waveform, spam base, ALOI, and Annthyroid are characterized by no clear separation in K-means and HDBSCAN. For K-means, this results in a uniform blending of outliers to inliers across clusters and K, the silhouette scores show little to no contrast. In HDBSCAN, outliers and inliers will be uniformly spread amongst one or few primary clusters wherein further decreases in eps result in very small inlier minority clusters and more inliers moving into the noise group.

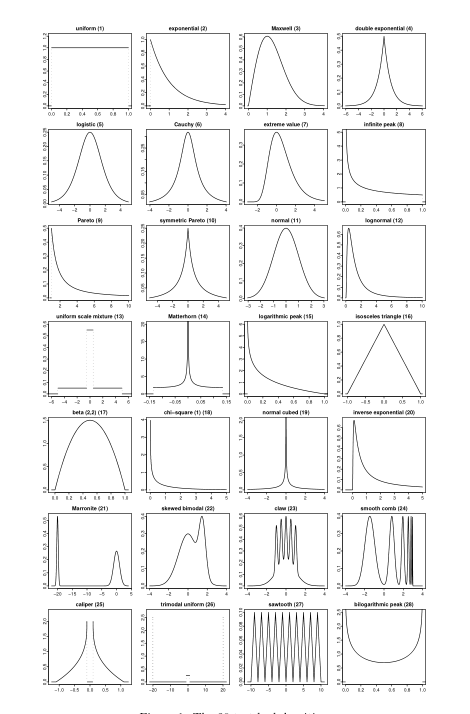


14 HDBSCAN Clustering Of ALOI

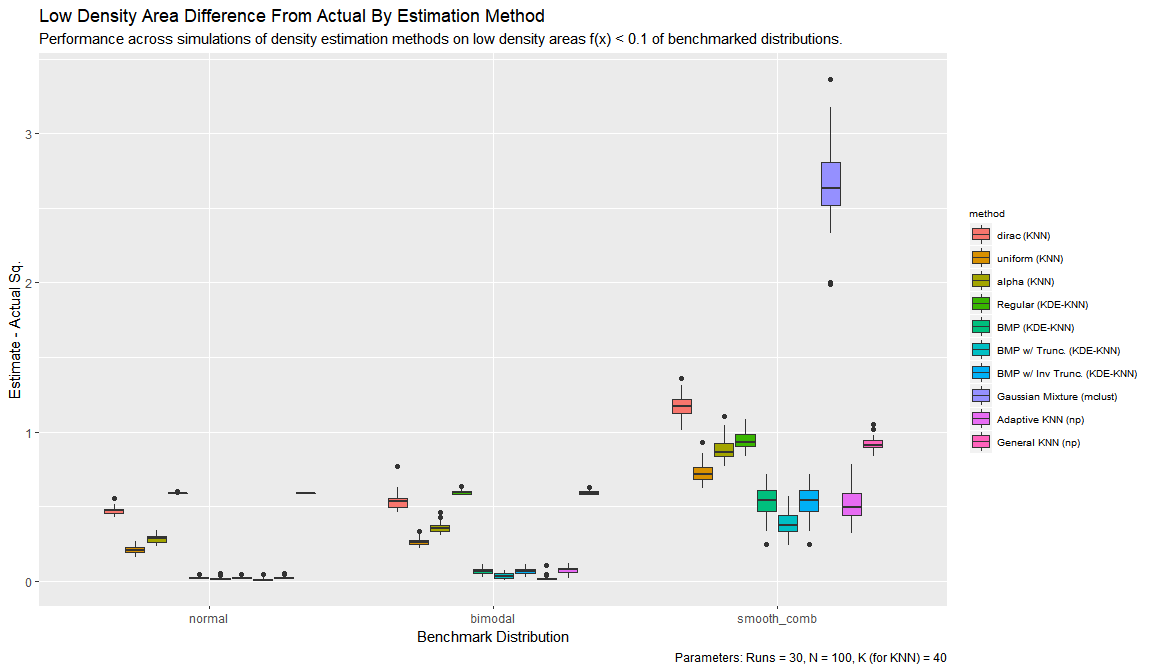


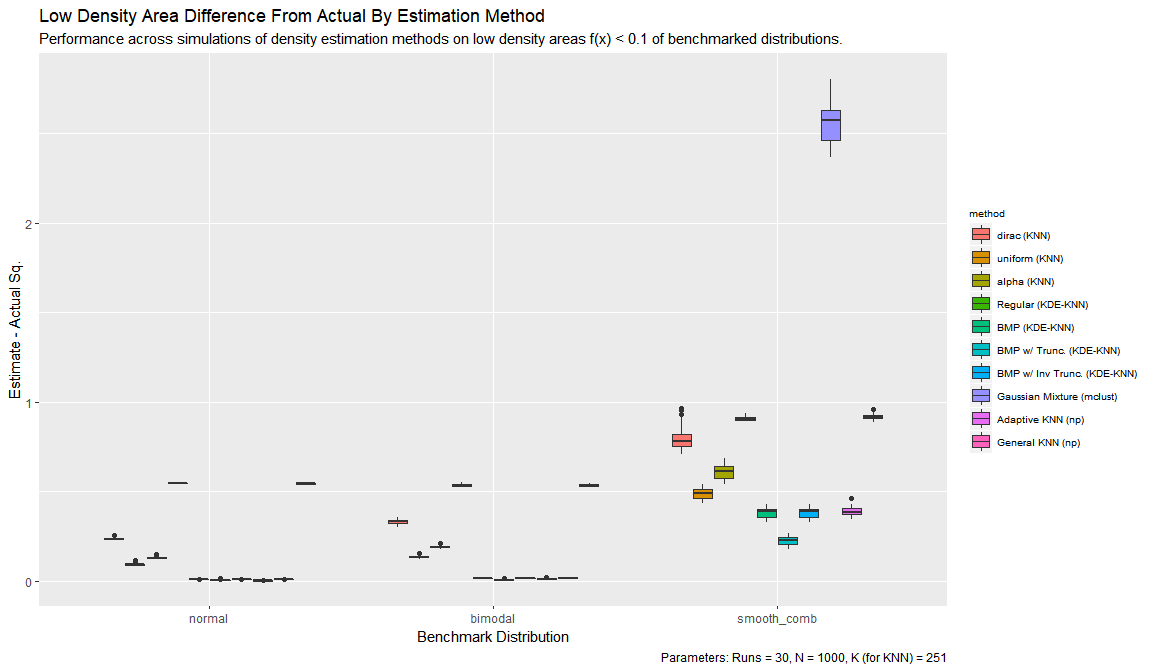
15 K-Means Clustering of ALOI

For HDBSCAN, often the initial clusterings (k<4) provided the best gauge of difficulty. The proportion of outliers to inliers in the initial noise grouping is often the highest. As eps decreases, significantly more inliers are marked as noise. Also, the formation of outlier clusters in HDBSCAN is a lot rarer than in K-means. Often, as eps decreases, to generate up to 15 clusters, the primary cluster gives off a relatively minor amount observations into either the noise group or very small clusters, less than 1% of n. In some datasets such as WBC, Arrhythmia, and Hepatitis, 15 clusters could not be generated, only 2 or 3 clusters could be formed. The silhouette scores of minority clusters are very strong compared to that of the primary cluster(s). The silhouette score of primary clusters(s) consistently decreases as eps drops and more clusters are introduced. The starting score of the primary cluster(s) varies, but across k they generally drop to near zero.

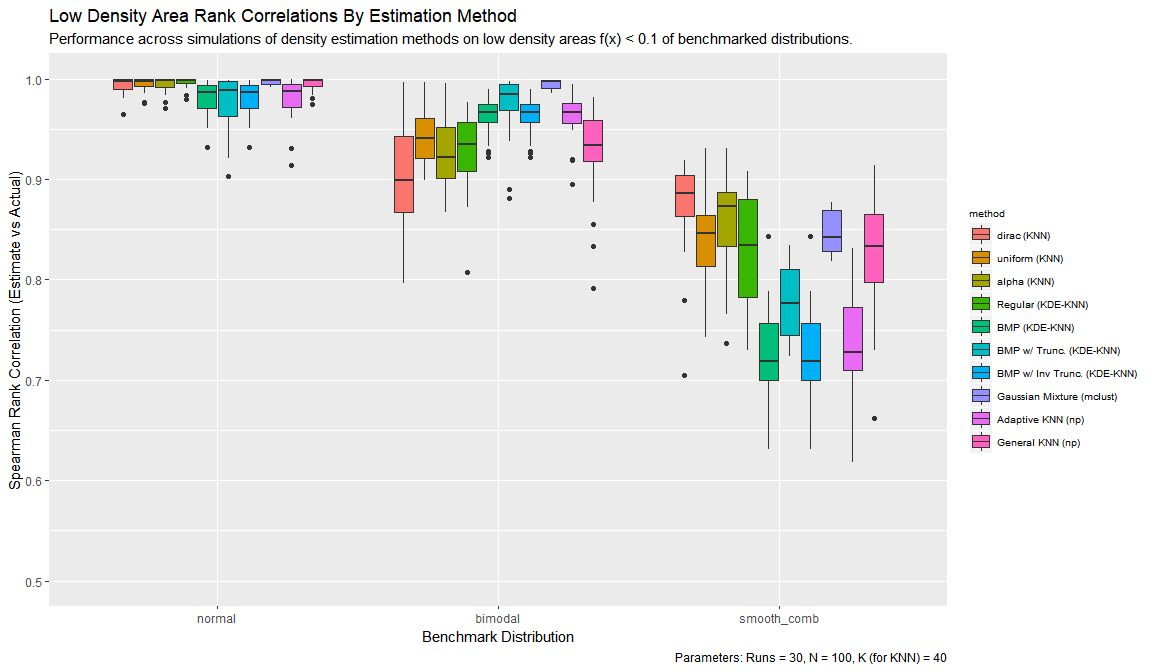


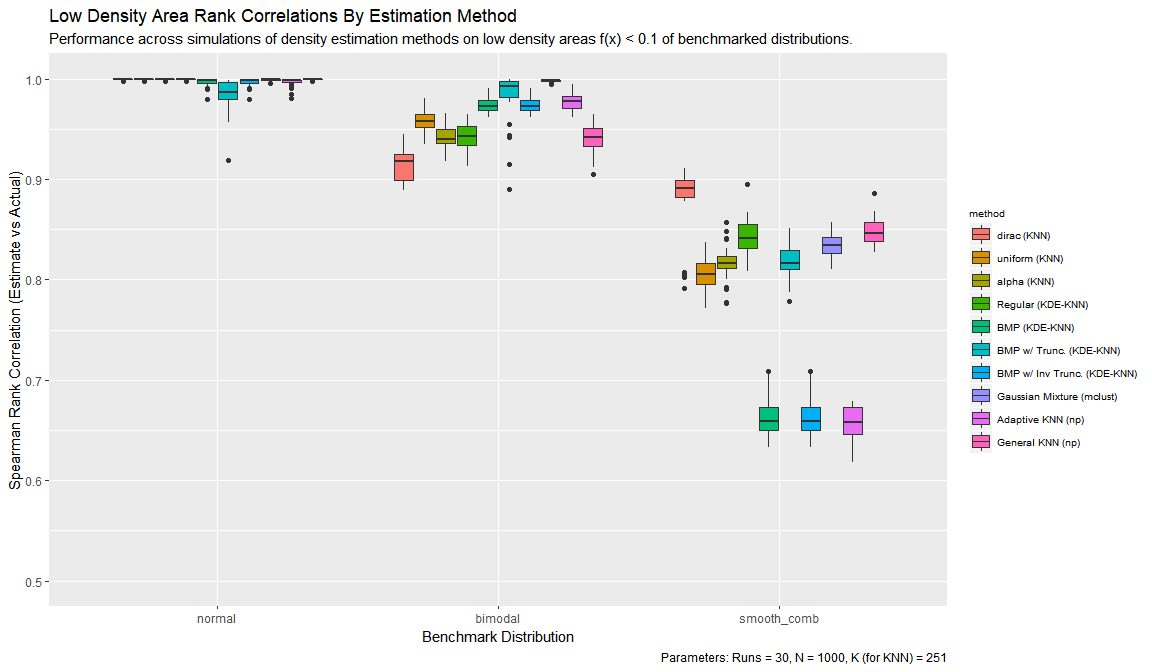
16 Appendix: Illustration of benchden densities.





17 Appendix: Summary of KDE Methods By Sum Of Difference From Actual





18 Appendix: Summary of KDE Methods By Rank Correlation Of Estimate To Actual