ABSTRACT

EXPLORING A STATISTICAL TEST FOR MODALITY

James E. Petkus, M.S.
Division of Statistics
Northern Illinois University, 2014
Alan Polansky, Director

This thesis explores the possibility of testing the statistical significance of modality based on the smoothed bootstrap test for modality. There is little information available on the plausibility of Silverman's test, potentially due to the computational requirements essential to perform the required bootstrap testing. Using the statistical package R, and incorporating parallel processing to perform large bootstrap simulations, the goal is to determine if a statistical test to estimate modality based on significance may exist. Simulation studies reveal properties of critical bandwidth and show modifications to the smoothed bootstrap test for modality make it possible to perform a statistical test for modality.

NORTHERN ILLINOIS UNIVERSITY
DEKALB, ILLINOIS

DECEMBER 2014

EXPLORING A STATISTICAL TEST FOR MODALITY

BY

JAMES E. PETKUS

A THESIS SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

DIVISION OF STATISTICS

Thesis Director:
Alan Polansky

ACKNOWLEDGEMENTS

I want to convey particular thanks to my thesis advisor, Dr. Alan Polansky, for providing unequivocal advice, encouragement, and placidity through the entire course of this thesis. I also want to thank the faculty and staff members in the Division of Statistics at Northern Illinois University for their knowledge, support, and patience; you all made my time in the degree program a challenging and enjoyable journey. A special thanks to the Department of Computer Science at Northern Illinois University for making available the high-performance computing facilities to students, which allowed great expansion to the statistical simulations. I would like to thank Kelly McPike for editing my thesis. Finally, thank you to all of my family and friends for your patience and tolerance over the past few years.

TABLE OF CONTENTS

Page

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

LIST OF APPENDICES

Appendix Page

CHAPTER 1

INTRODUCTION

1.1 Interest in Modality

A mode is a local maximum found in a density function, and modality is the number of modes counted in the same density function (Silverman, 1986). Many fields of study, especially those of a scientific nature, are interested in the presence of multiple modes, or multimodal density functions, because it is a probable indicator that a mixture of components exists (Cox, 1966). This idea stems from cluster analysis, in which the objective is to identify differences that may be present in a collection of data. Suitable examples include the differences between species or gender, mineral composition or fracture orientation in rocks, animal or insect activities, and weather or climate information. Good and Gaskins (1980) comment on modality used in high-energy physics to provide evidence of elementary particles or as an indicating feature of a random variable.

One area of study where the number of modes is of particular interest is the study of how water disseminates through fracture networks in the rock layers. Geologists use modality in the calculations used to determine the percolation and transport properties of these fractures (Ekneligoda & Henkel, 2010; Manzocchi, 2002). However, it is important to note much of the data found in the geological sciences is circular data, and most tests for modality are for linear data. Circular data is somewhat different than linear data because the information is collected

based on orientation, direction or angle, and it may include time-series or cyclic periods, not just a point or vector. Fisher (1993) proposes an adapted methodology based on Silverman's (1986) tests for modality on linear data, noting that it is often difficult to determine the modality of data with complex directional components.

1.2 Distribution Visualization

Circular data typically represents a vector, axis of directional orientation, or some angular measurement. Understanding the data and what it represents is important before starting any analysis. Visualization of the information will help to determine important characteristics that may be contained within the data. This may influence the decisions on model selection, forecasting, and modality (Fisher, 1993).

Histograms are elementary in construction and provide a practical way to visualize the information. Circular data is easy to display on a linear histogram; however, the angular component is lost in the visualization. Plotting circular data onto a linear histogram, with frequency on the y-axis and angle on the x-axis, makes visualizing the angular component difficult. Conversely, plotting the same circular data onto an angular histogram, the directional component becomes much more obvious (Fisher, 1993). Using information collected on the orientations of termite mounds (Appendix A.1 & A.2), the difference between the two histograms is perceptibly evident between the linear histogram (Figure 1) and the circular histogram (Figure 2).

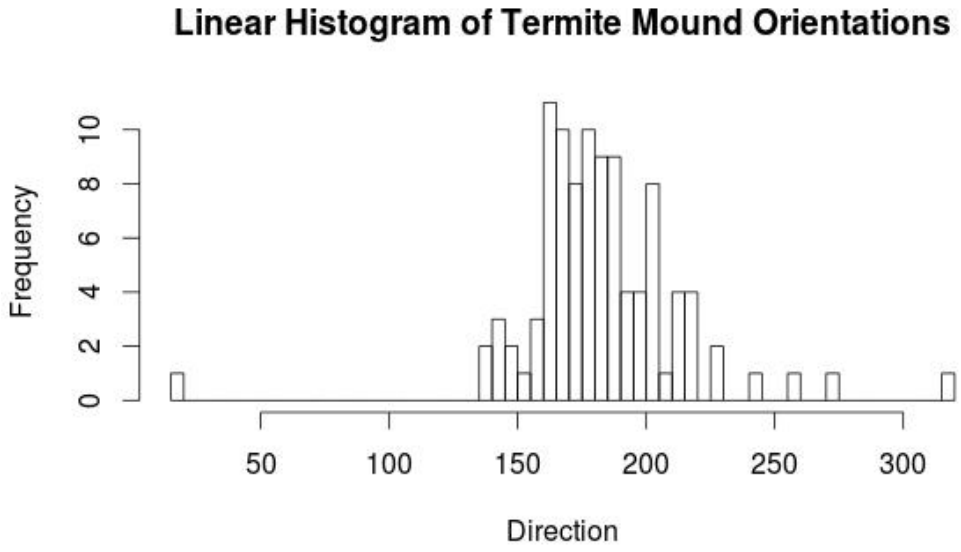

Figure 1: Linear histogram for the orientation of termite mounds (Appendix A.1).

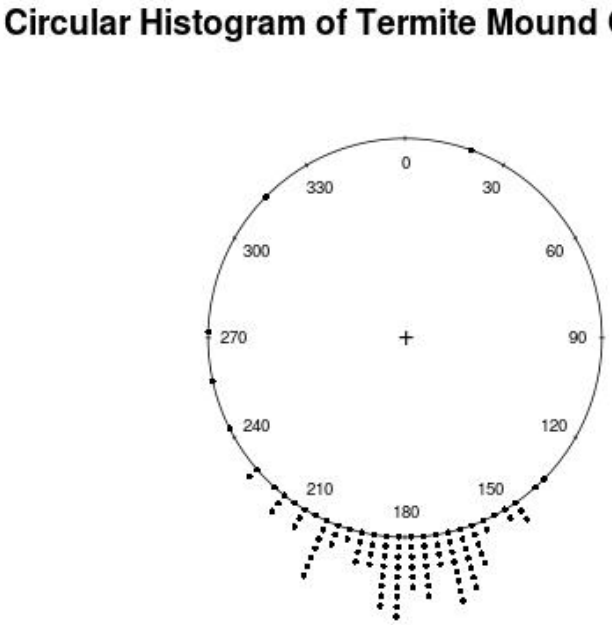

Figure 2: Circular histogram for the orientation of termite mounds (Appendix A.2).

The linear histogram of the data appears strongly unimodal and positively skewed, yet it has some evidence of belonging to a normal distribution. However, the circular histogram portrays a different picture, with two or three prominent peaks in the data, and looks less like it belongs to a normal distribution. This simple example demonstrates how assumptions about the modality can differ based on the visual representation of the data. Additional visualization is necessary to improve assumptions about the data.

1.3 Density Visualization

Visualization of the histogram alone does not provide enough information about the data; this necessitates a function to describe the data. However, the true distribution of the population remains unknown. Estimation of the probability density function using the data is one way to overcome this. Because this data is non-parametric, kernel density estimation can be used to create a representative function. This estimation is based on an assumed underlying distribution function, referred to as the kernel. The kernel function is positioned at each point, or groups of points, on the histogram (Figure 3). Each kernel function contributes to the overall function; where the kernel functions overlap, the sum of each contribution is used (Figure 4). A new function to represent the overall density function for the data is now constructed; this is the kernel density estimate (Silverman, 1986).

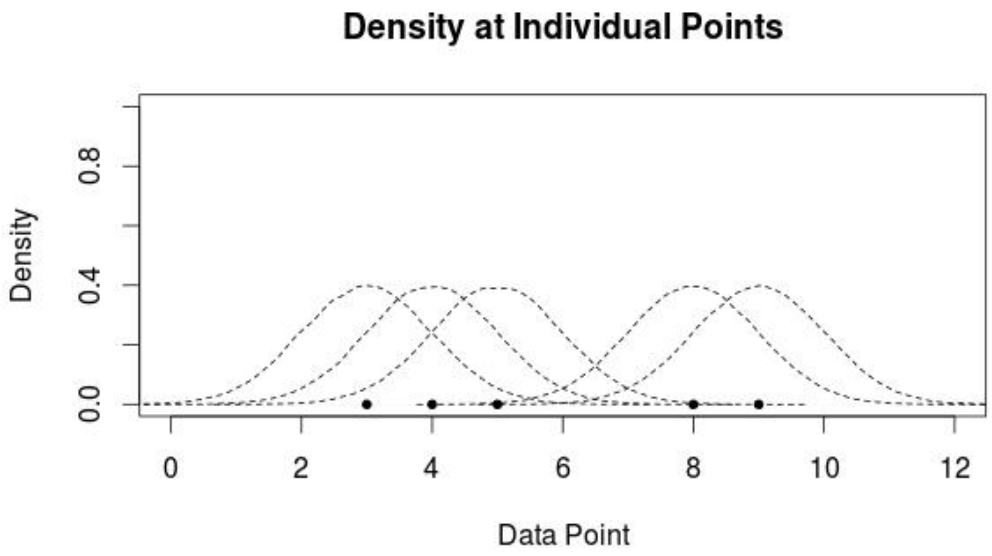

Figure 3: Kernel density function positioned about each point.

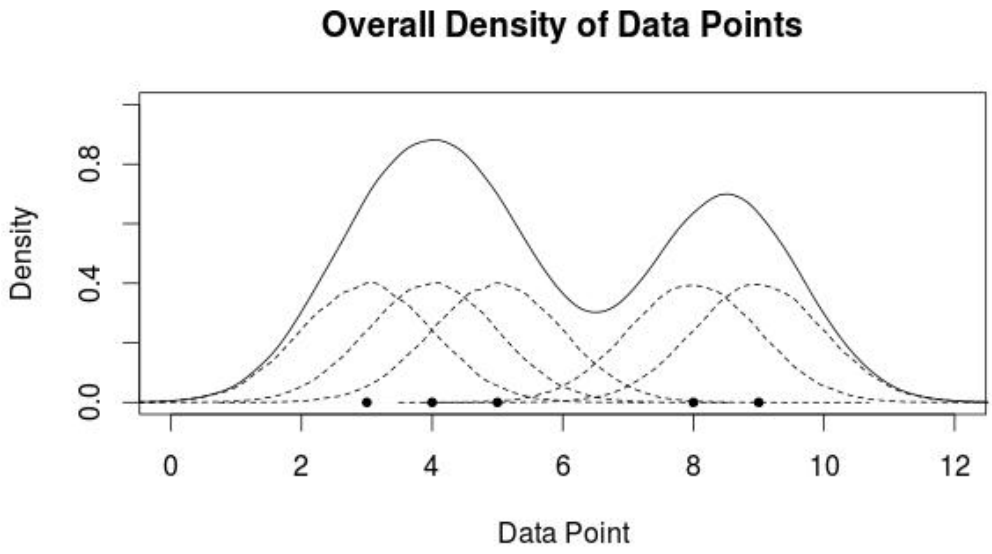

Figure 4: Resulting density estimate function created from kernel density functions.

This example of kernel density estimate assumes the kernel follows a normal density at each data point. While any other density function can be used for the kernel, the normal density tends to be the most commonly used kernel in density estimating functions. Using the histogram and the density function, a visual representation of the distribution can be created for both the linear data (Figure 5) and the circular data (Figure 6). This is the starting point for estimating the modality of the data.

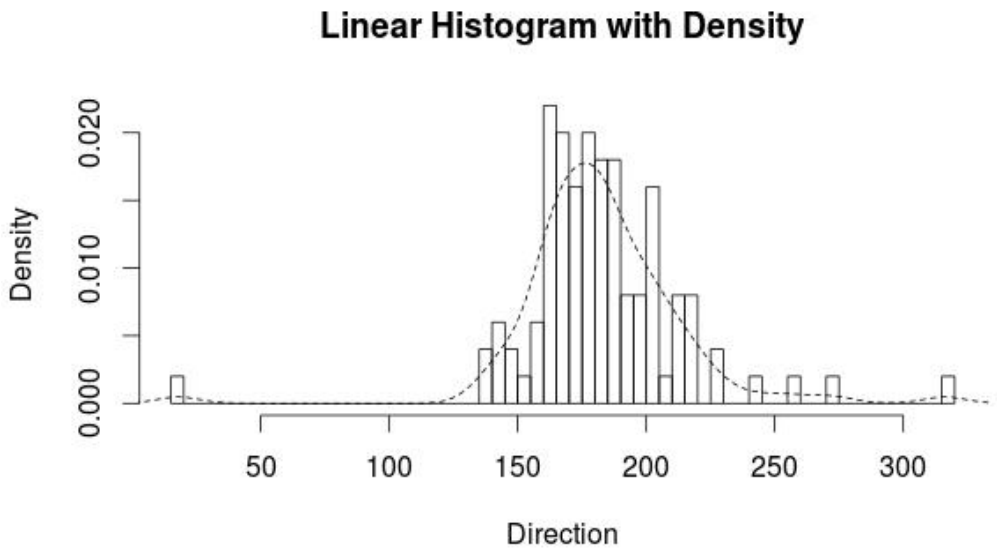

Figure 5: Linear histogram with unimodal density estimate for termite mound.

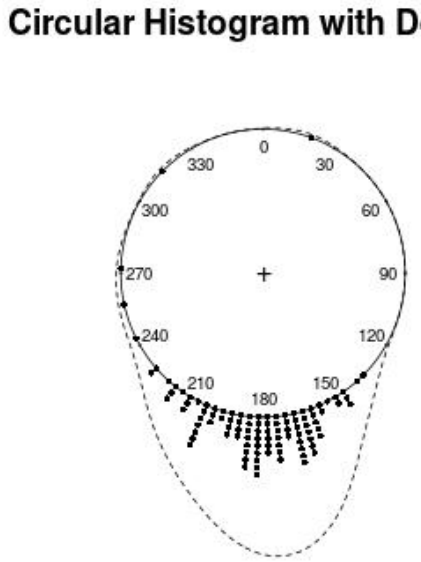

Figure 6: Circular histogram with unimodal density estimate for termite mounds.

With the density functions added to the histograms, previous observations regarding the modality of the data change. Although it seems mostly unimodal, the linear histogram appears to have the potential for three modes with the density function added. Although the circular histogram appears to have two or three modes, the added density function indicates it is more unimodal. The reason for this is that the density function, with no parameters defined, uses the best fitting density estimation. Parameters for the density function are usually size, equal to the number of data points, and bandwidth, a value producing a smoother density function. Values that are larger produce smoother density curves, less modes; values that are smaller produce bumpy density curves, more modes (Jones, 1983). It is important to note the choice of a smoothing value is largely arbitrary with several different methods of suggested selection, which

makes the parameter insufficient to use when estimating modality. A solid method to evaluate the statistical significance of modality does not currently exist (Silverman, 1986).

This thesis will explore the possibility of testing the statistical significance based on Silverman's smoothed bootstrap test for multimodality (Silverman, 1986). There is little information available on the plausibility of Silverman's test, likely due to the computing power needed to perform the bootstrap testing necessary. Using the statistical programming language R, and incorporating parallel processing to perform large bootstraps, the goal is to determine if a statistical test for significance may exist.

Chapter 2 elaborates on the theoretical basis used to explore, analyze, and test modality. It will include the kernel density estimating function for both linear and circular densities, an improved explanation of how bandwidth works, and the procedures for Silverman's (1986) test for modality. Chapter 3 contains the exploration and analysis of modality by applying Silverman's (1986) test to actual data. Evidence suggests obstacles in the process, indicating a new or adapted methodology may be required. Additionally, there are some interesting properties of bandwidth revealed in the resulting simulation studies. Chapter 4 performs modality testing on actual data using the adapted methodology from Chapter 3. Last, the thesis will discuss conclusions derived from the research, followed by recommendations for future research.

CHAPTER 2

THEORETICAL BASIS

2.1 Kernel Density Estimating Function

The default density estimating function in R assumes the normal density and uses the basic kernel estimating function:

$$\hat{f}_{kde}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right) \tag{2.1}$$

where *n* is the length of the data, *h* is the bandwidth employed to determine the amount of smoothing to be utilized, *x* represents the value at each data point, and *K* is the kernel density function applied to each group of points that will be included in the smoothing. The kernel density function should be a symmetric function that has a total integration value of one. Realize that the behavior of the overall kernel density estimate relies heavily on the choice of *h*, which is generally an arbitrary method of selection. Large values of *h* will produce smoother functions with less modes. Additionally, *n* must be large enough to estimate the kernel density function with sufficient information and will require more smoothing (Silverman, 1986).

Fisher (1993) expands these methods, applies them to circular-based data, and uses the basic kernel estimating function:

$$\hat{f}_{ckde}(\theta) = \frac{1}{nh}\sum_{i=1}^{n} w\left(\frac{\theta-\theta_i}{h}\right) \quad (2.2)$$

where *n* and *h* remain the same as above, θ is the angular value of each data point, and *w* is the circular kernel density function about each point. It is important to understand the smoothing parameter *h* (bandwidth) operates in reverse when applied to circular data. This means smaller values of *h* will produce smoother functions with less modes (Fisher, 1993).

Because the selection of *h* is decidedly arbitrary, it is possible to select a value for *h* that is larger than desired for linear applications. Large values of *h* are required to smooth data that is multimodal in nature. There is a way to evaluate *h* against a standard family of unimodal densities as indicated by simulation studies conducted by (Jones, 1983). The results of these studies show samples drawn from normal distribution have an estimated proportion of results that are multimodal (Table 1).

Table 1

Estimate of samples yielding multimodal results for *h*.

h	Proportion
$1.25\sigma n^{-1/5}$	5%
$1.06\sigma n^{-1/5}$	15%
$0.95\sigma n^{-1/5}$	33%

σ is the standard deviation of the data and *n* is the sample size equal to the length of data.

According to the study, these results apply to values of n, where $40 \leq n \leq 5000$ (Silverman, 1986). Mathematical proofs and prior simulation studies demonstrate similar results for circular data (Taylor, 2008).

2.2 Silverman's Test for Modality

Silverman (1986) proposes the construction of a test for multimodality using a smoothed bootstrap test based on critical smoothing. The idea of arbitrarily selecting a value for bandwidth and fitting it to the histogram is problematic. Critical smoothing removes the notion of arbitrary or optimum bandwidth selection by using the behavioral relationship between bandwidth and the number of modes found. Holding all other variables constant and decreasing bandwidth, there will come a point where the number of modes found increases. The bandwidth just before the point where the mode count increases is the critical bandwidth, and it can be calculated to a reasonable precision (Silverman, 1986).

A smoothed bootstrap approach constructs simulations similar to the standard bootstrap method, but draws samples from a smoothed kernel density as opposed to sampling from the observed data. This approach eliminates the undesired effect of samples containing repeat values from the observed data, which allows an improved estimate of the distribution function to which the data may belong. Additionally, the smoothed bootstrap method also allows for a more nonparametric approach (Silverman, 1986). Simulation studies reveal a substantial improvement in the root mean squared error using the smoothed bootstrap compared the standard bootstrap (Efron, 1981). However, the selection of an adequate smoothing parameter and a lack of

systematic investigation, into conditions under which improved results occur, still remain (Silverman, 1986).

It is possible to construct a test for multimodality using the critical smoothing method to select a bandwidth for the smoothed bootstrap approach. By comparing the critical bandwidth to a bandwidth obtained from a suitable unimodal density function, it can be determined whether the data points reasonably belong to the unimodal density function. A suitable density function with which to test for unimodality will have the following desirable properties (Silverman, 1986):

1. "The density function must be unimodal, since the density function must be a representative of the compound null hypothesis of unimodality" (Silverman, 1986, p. 139).
2. "Subject to (1), the density function should be a plausible density underlying the data; testing against all possible unimodal densities is a hopeless task, since, for example, large values of critical bandwidth would be obtained from unimodal densities with very large variances" (Silverman, 1986, p. 139).
3. "In order to give unimodality a fair chance of explaining the data, density function should be, in some sense, the most nearly bimodal among those densities satisfying (1) and (2)" (Silverman, 1986, p. 139).

These conditions can be satisfied by setting the smoothed bootstrap density estimate equal to the density estimate constructed using the critical bandwidth found for the unimodal

case. A density estimate constructed using the critical bandwidth is an extreme unimodal density because reducing the bandwidth any further will make the density function multimodal. Taking a large number of samples, with a size equal to that of the data, from the extreme unimodal density function, the proportion of samples that have bandwidths greater than the critical bandwidth can be calculated. This may be simplified by obtaining the mode count for each sample using the critical bandwidth and calculating the percentage of samples that yield multimodal results. The proportion calculated using either method is the p-value of the bandwidth (Silverman, 1986).

Silverman's test for modality, summarized:

1. Given a dataset $X_1, \ldots, X_n$, find h_0, the critical bandwidth for the unimodal case.
2. Find f_0, the extreme unimodal density function of the data with bandwidth h_0.
3. Generate sample of size n from f_0 and calculate the modality using bandwidth h_0.
4. Repeat (3) a large number of times, B.
5. Find p-value, the proportion of samples that yield multimodality.

CHAPTER 3

MODALITY ANALYSIS

3.1 Applying Silverman's Test

Using the methodology outlined by Silverman (1986) and adapted to circular data by Fisher (1983), testing of the methodology takes place on geological information. This initial test uses a dataset containing azimuth measurements found in rock layers, provided by the Geology Department at NIU (Appendix A.3). The data has three sets of measurements; each set contains a count of cracks associated with azimuth angles from 1 to 360. Set A, containing 4230 observations, is the primary dataset that will be used to conduct the initial application of modality testing.

The first step is to find the critical bandwidth for the unimodal case. This is done using a routine written in R (Appendix B.1) that applies Silverman's method of incrementing or decrementing the bandwidth until the desired level of precision is reached (Silverman, 1986). The operating density function is the ‘rvonmisis’ circular density routine found in the R package 'circular' and uses the equation by Fisher (1983). The resulting critical bandwidth is 203.6821 for the extreme unimodal case of this data. A circular histogram provides visualization of the data and resulting density function at the critical bandwidth (Figure 7).

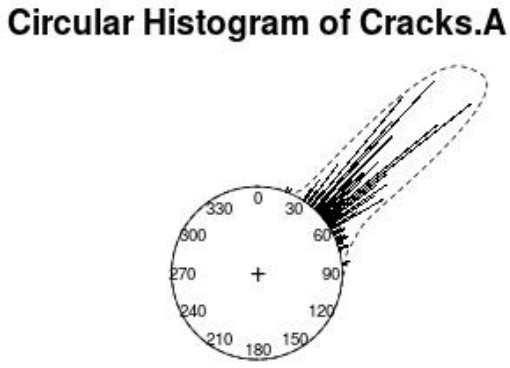

Figure 7: Circular histogram of azimuths with extreme unimodal density function.

A unimodal density function for this data, step 2 in the test, is generated using the circular density routine in R with a bandwidth equal to the result. Samples are now generated from the resulting extreme unimodal density. The mode for each sample is calculated using the critical bandwidth obtained from the data. In this case $B = 1000$ samples were generated from the extreme unimodal density using a routine written in R (Appendix B.2). The last step is to calculate the proportion of samples yielding multimodal results, performed by the same R routine that generates the samples (Appendix B.2). The proportion of results found to be multimodal, in this case bimodal, is 75%.

This is not the anticipated result based on the studies conducted by Jones (1983). The expectation is that there will be less results that are multimodal as the bandwidth gets closer to the point where it becomes the extreme bandwidth. Data generated using the ‘rvonmisis’ routine

from the R package 'circular' is compared to the results from the azimuth data. Test results conducted on $B = 1000$ datasets generated using this method, with an observation count equal to 4230, returned proportions between 90% and 100%. Noting that tests by Silverman (1986) and Fisher (1993), were performed using chondrite meteor data (Appendix A.2), the same data is used here for an additional evaluation. The proportion of results found to be multimodal in the meteor data is 40%. Notably, the results are not matching up, leaving two possible conclusions. First, there is an error with the routine coded in R, and secondly, there is an oversight in the proposed method. Under the assumption that the routine coded in R is erroneous due to attempting the tests in circular space, it is best to transition to a linear space and perform simulation tests again.

3.2 Simulation Tests for Modality Proportions

In order to perform the simulation tests within a linear reference, the routines coded in R for circular data must be re-coded. The resulting R routines that will calculate bandwidth, generate samples, count modes, and calculate proportions are added (Appendix B.4, B.5, and B.6). Jones (1983) provides information about the percentage of samples that should be multimodal under specific bandwidth calculations; however, this is of little use when working with the extreme critical bandwidth between unimodal and multimodal. A routine coded in R that will test an array of values for both data size and standard deviation (Appendix B.7) will return the percentage of multimodal results. The test generates $B = 1000$ samples from each combination of data size and standard deviation using the ‘rnorm’ routine with a fixed mean (see

Table 2). Arrays for data size and standard deviation are respectively: *c(50, 100, 500, 1000, 5000, 10000)* and *c(0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0)*.

Table 2

Proportion of multimodal results for each *n* and *sd* combination.

n \ sd	*0.500*	*0.600*	*0.700*	*0.800*	*0.900*	*1.000*	*2.000*	*3.000*
50	0.063	0.074	0.078	0.081	0.069	0.100	0.167	0.191
100	0.023	0.039	0.046	0.037	0.042	0.035	0.078	0.096
500	0.005	0.005	0.009	0.003	0.004	0.008	0.006	0.014
1000	0.001	0.001	0.004	0.003	0.002	0.002	0.006	0.009
5000	0.000	0.000	0.001	0.002	0.000	0.000	0.001	0.003
10000	0.002	0.000	0.000	0.001	0.000	0.000	0.002	0.001

Results for the simulation reveal that larger data size or smaller standard deviation tend to decrease the proportion of results that are multimodal at the critical bandwidth. This is expected as we are drawing samples from a known unimodal density function. Additionally, the larger the size the closer it represents the distribution from which it is drawn. If there are few data points, or the data points are farther apart, more smoothing is required to obtain a unimodal density function (Silverman, 1986).

Silverman's (1986) method of drawing samples from the extreme density function will tend to yield a higher proportion of multimodal results when the data size is large because the extreme density function is nearly bimodal. Jones (1983) suggests that setting the bandwidth to $1.25\sigma n^{-1/5}$ will yield multimodal results in approximately 5% of samples. Using the same R code from the previous simulation with a fixed bandwidth as suggested and adjusted arrays for both data size, *c(50, 100, 500, 1000, 5000)*, and standard deviation, *c(0.01, 0.05, 0.10, 0.50, 1.0, 2.0, 3.0)*, the proportion of each combination over the suggested bandwidth is calculated (Table 3).

Table 3

Proportion of multimodal results for $h = 1.25\sigma n^{-1/5}$ (expectation 5%).

n \ sd	*0.010*	*0.050*	*0.100*	*0.500*	*1.000*	*2.000*	*3.000*
50	0.039	0.033	0.037	0.029	0.042	0.032	0.034
100	0.044	0.049	0.041	0.044	0.040	0.042	0.049
500	0.089	0.084	0.097	0.079	0.102	0.098	0.109
1000	0.131	0.136	0.138	0.130	.0144	0.143	0.138
5000	0.283	0.253	0.277	0.263	0.274	0.258	0.261

The results show standard deviation appears to have little effect on the proportion. However, as the data size increases, the proportion of multimodal results also increases. Once the data size reaches a value somewhere over 100, the number of multimodal results is always over

the 5% threshold proposed and continues to increase with the data size. Similar results occur when setting the bandwidth to $1.06\sigma n^{-1/5}$ and $0.95\sigma n^{-1/5}$, with expected multimodal results of 15% and 33%, respectively, from the samples (Tables 4 & 5).

Table 4

Proportion of multimodal results for $h = 1.06\sigma n^{-1/5}$ (expectation 15%).

n \ sd	*0.010*	*0.050*	*0.100*	*0.500*	*1.000*	*2.000*	*3.000*
50	0.130	0.136	0.138	0.120	0.133	0.124	0.136
100	0.129	0.118	0.153	0.125	0.124	0.136	0.130
500	0.233	0.196	0.190	0.217	0.179	0.217	0.181
1000	0.254	0.242	0.249	0.268	0.257	0.257	0.236
5000	0.436	0.442	0.458	0.416	0.454	0.417	0.423

Table 5

Proportion of multimodal results for $h = 0.95\sigma n^{-1/5}$ (expectation 33%).

n / sd	*0.010*	*0.050*	*0.100*	*0.500*	*1.000*	*2.000*	*3.000*
50	0.333	0.337	0.305	0.298	0.306	0.349	0.304
100	0.307	0.291	0.321	0.297	0.301	0.288	0.311
500	0.387	0.412	0.420	0.416	0.379	0.412	0.376
1000	0.447	0.457	0.468	0.454	0.469	0.470	0.454
5000	0.671	0.682	0.644	0.651	0.639	0.627	0.664

This simulation study indicates potential issues in the outlined methodology for testing the modality of a dataset. Drawing samples from the extreme density function, created using the data, is not necessarily representative of how modality behaves in all situations. In effect, this is testing the modality of the data against the data itself, when it should be tested against the behavior of data drawn from a known modal distribution. Further, the test for modality will only work for a narrow range of data sizes, which is a very restrictive factor.

3.3 Modifications to Silverman's Test

The first problem to overcome is drawing samples from the extreme density function. It may be more appropriate to compare the critical bandwidth obtained from the data against critical bandwidths calculated from a large number of samples, where samples are generated

from a density function with the same modality and parameters as the data. For example, in the unimodal case, draw samples from a known unimodal distribution, such as the normal distribution, with a data size and standard deviation equal to that of the data. Next, find the extreme critical bandwidth for each sample at the unimodal boundary. Now, compare the critical bandwidth from the data to the distribution of bandwidths obtained from the simulation.

The sample datasets for chondrite meteors and crack azimuths are used to demonstrate proposed modifications to the modality test. The datasets are converted first to a linear frame of reference, which does not alter the data (Figures 8 & 9). However, it is important to note, linear modality testing may not have the same results as circular modality testing. The extreme unimodal critical bandwidths for the datasets are 2.3959 for meteors and 4.0259 for azimuths, found using the R routine, which locates the critical bandwidth between two modes (Appendix B.4).

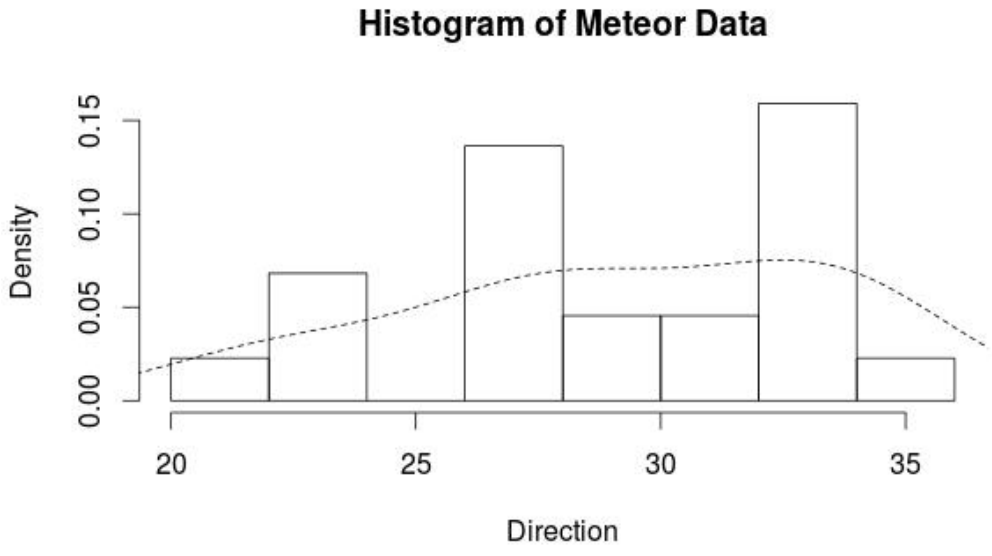

Figure 8: Linear histogram of meteor data with extreme unimodal density function.

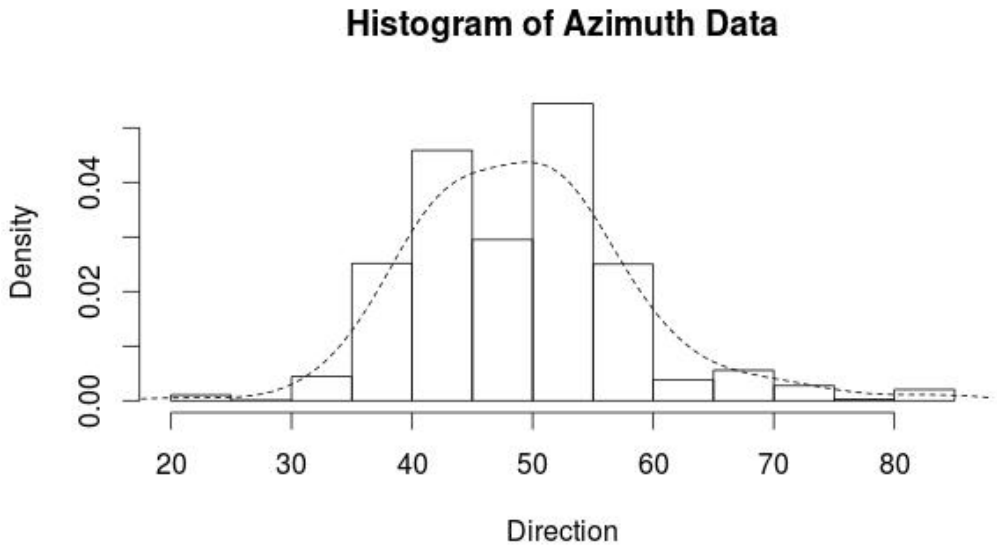

Figure 9: Linear histogram of azimuth data with extreme unimodal density function.

Using parameters obtained from the datasets, $B = 1000$ samples are generated from the standard normal for each dataset. The meteor dataset has a standard deviation of 4.2915 and a data size of 22. The azimuth dataset has a standard deviation of 8.8830 and a data size of 4230. Critical bandwidth results are returned by the R routine, which generates samples and calculates the bandwidth of each sample (Figures 10 & 11; Appendix B.5).

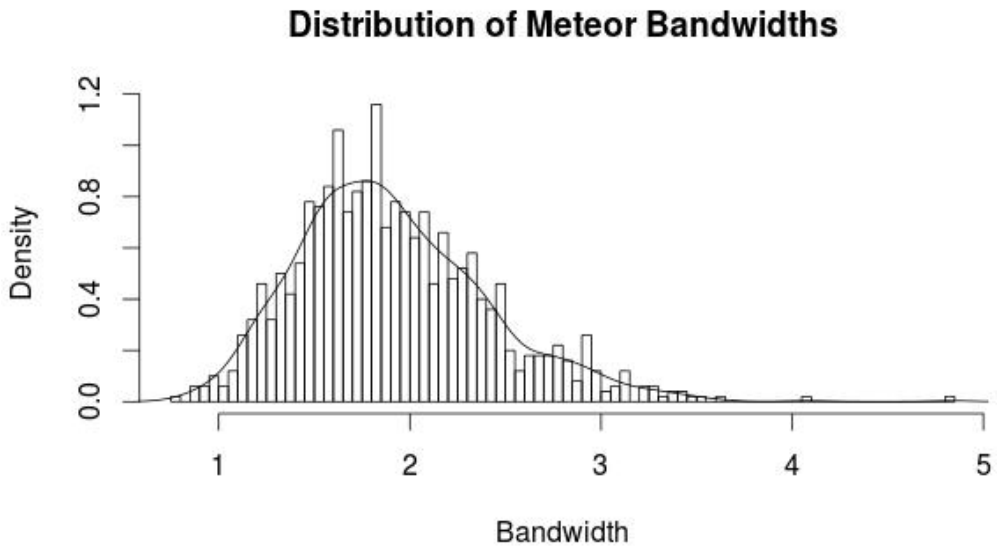

Figure 10: Distribution of critical bandwidths obtained from samples generated using normal distribution with parameters matching meteor data.

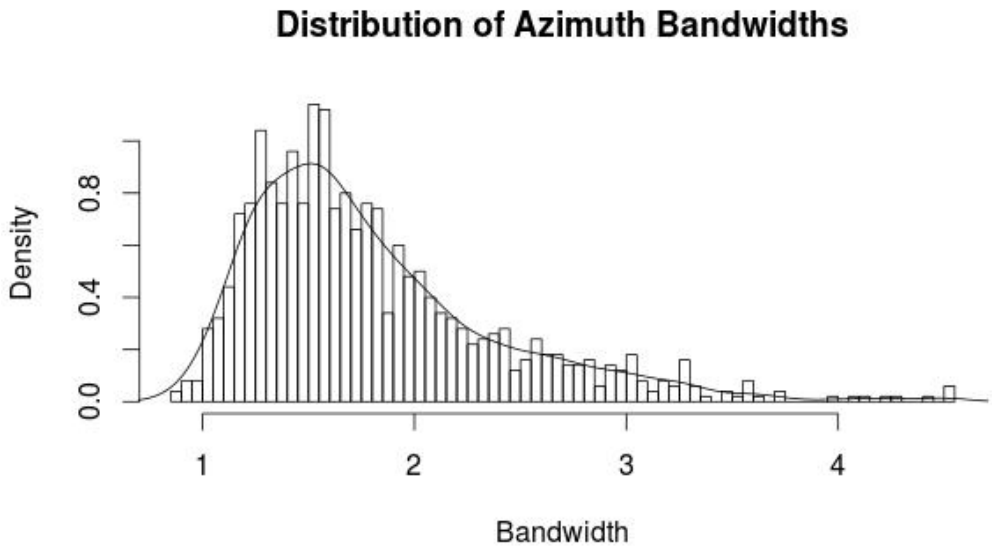

Figure 11: Distribution of critical bandwidths obtained from samples generated using normal distribution with parameters matching azimuth data.

The critical bandwidth for the data can now be compared to the distribution of bandwidths for any proportion desired. This proportion is the p-value against which testing can be done. Thus, setting the p-value to 5% and comparing the results show that the critical bandwidth for the meteor dataset falls well under the 5% threshold and may be unimodal (Figure 12). However, the critical bandwidth for azimuth data is beyond the 5% threshold and may not be unimodal (Figure 13).

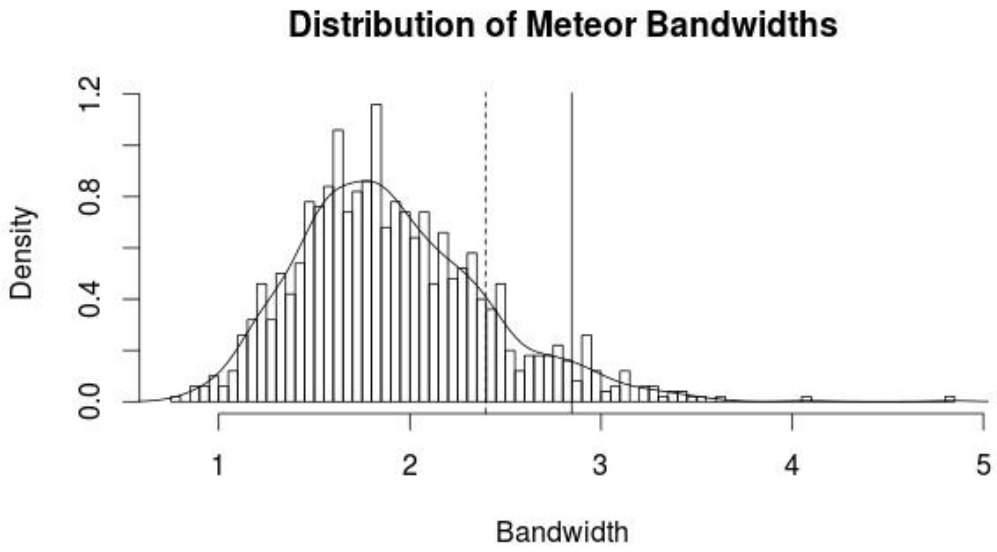

Figure 12: Critical bandwidth of meteor data (dashed line) and p-value of 5% (solid line).

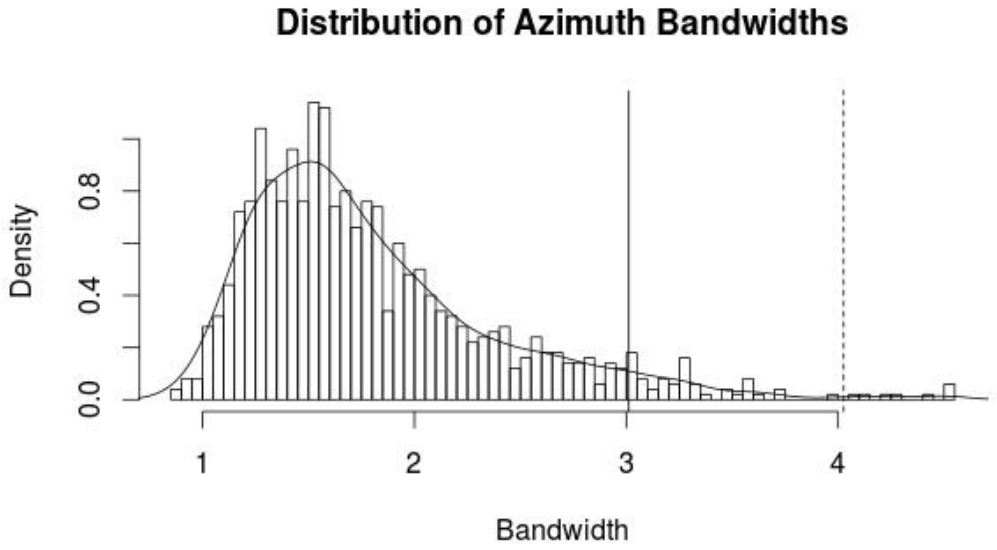

Figure 13: Critical bandwidth of azimuth data (dashed line) and p-value of 5% (solid line).

Both problems, noted in Section 3.2, appear to be corrected by this modification. Samples, no longer drawn from the dataset, are now independent of the data and represent a better range of possibilities for bandwidth. Size of the dataset is also no longer an issue; the distribution of the bandwidth results adjust based on the parameters of the data. Generating a number of samples even larger makes it possible to determine the p-value specific to a dataset more accurately. However, a new problem is now evident, large dataset sizes and sample counts will increase the computational requirements dramatically. Additionally, every dataset involves variable parameters and independent bootstraps. There may be properties exhibited by the distribution of bandwidths that can help to reduce the time and complexity of determining modality.

3.4 Properties of Critical Bandwidth

The density function, from which samples are drawn, is created based on parameters obtained from the dataset. In the unimodal case, a normal density is used, with a standard deviation equal to that of the dataset and a mean equal to zero since it does not affect the bandwidth calculations. By using a normal kernel density function, the assumption is that the various modes found in the data follow a normal density. Under this assumption, a location-scale adjustment can be made to the data so the standard deviation is always equal to 1. This will not affect the resulting critical density function, but it will change the bandwidth results (Figure 14).

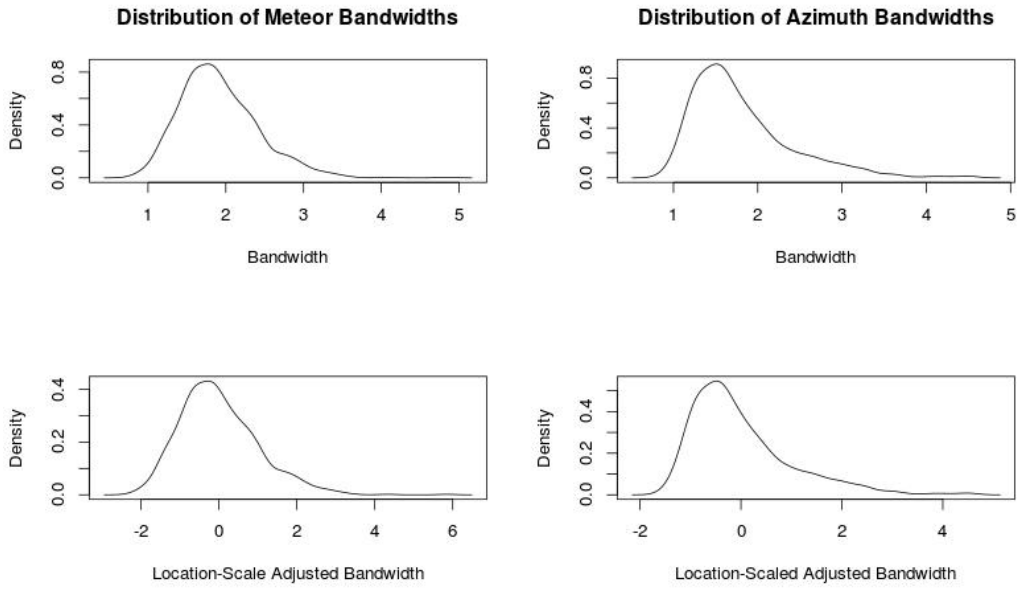

Figure 14: Distribution of critical bandwidths: before and after location-scale adjustment.

The number of combinations necessary to discern potential properties of bandwidth is greatly reduced. By performing a simulation study of resulting bandwidths obtained from samples generated with various data sizes, all with a standard deviation equal to 1, other potential properties are revealed (Figure 15).

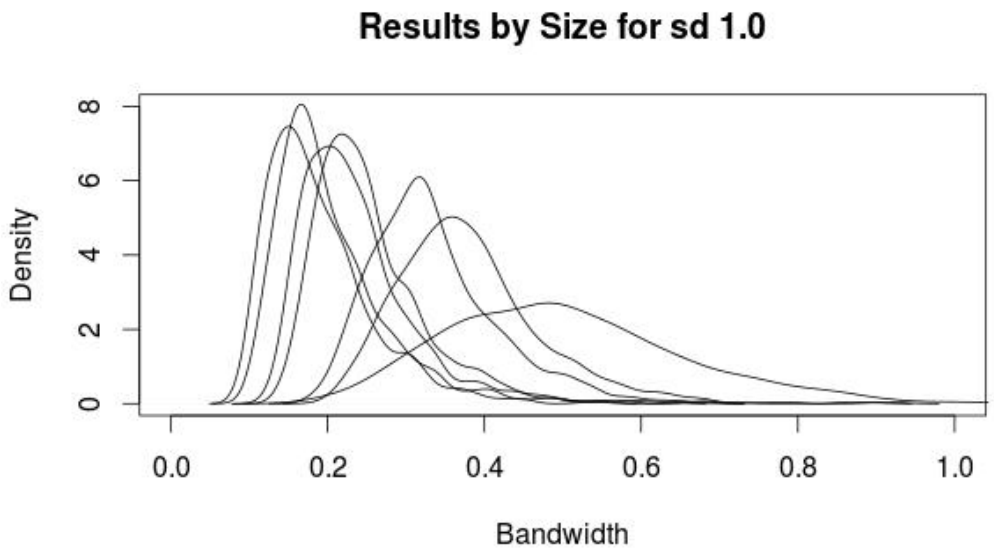

Figure 15: Resulting distributions as size changes with other parameters fixed.

Data sizes of resulting distributions here, from left to right, are 10000, 5000, 1000, 500, 100, 50, 10. The distribution of bandwidths found in $B = 1000$ samples of each size displays a shift towards higher bandwidths as size decreases. A similarity emerges by applying a location-scale adjustment to the bandwidth distributions (Figure 16).

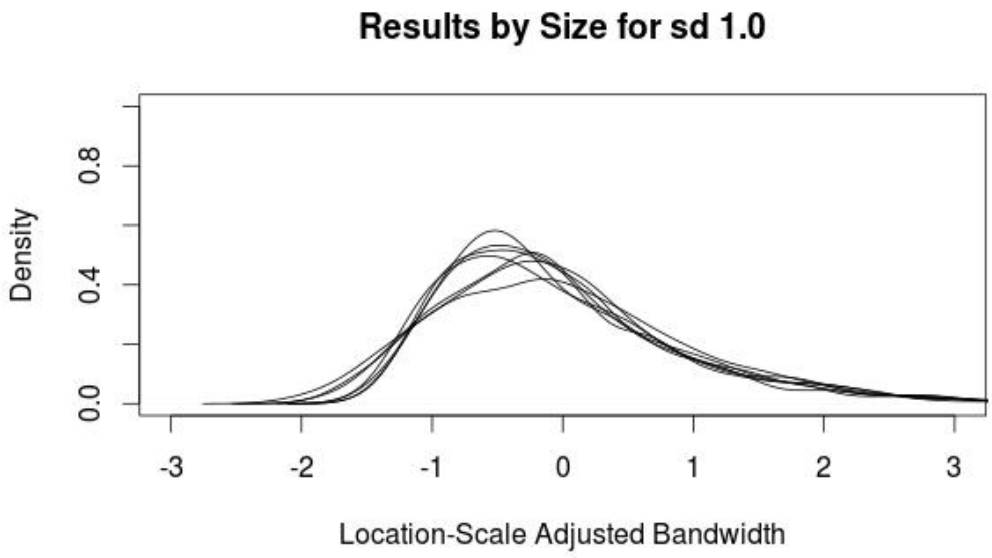

Figure 16: Resulting distributions from Figure 15 with location-scale adjustment.

The distribution of bandwidths appears to be following similar distribution patterns, possibly from the gamma family of distributions. By increasing the number of samples and the number of data sizes simulated for gathering bandwidths, a better representation is available (Figures 17 & 18).

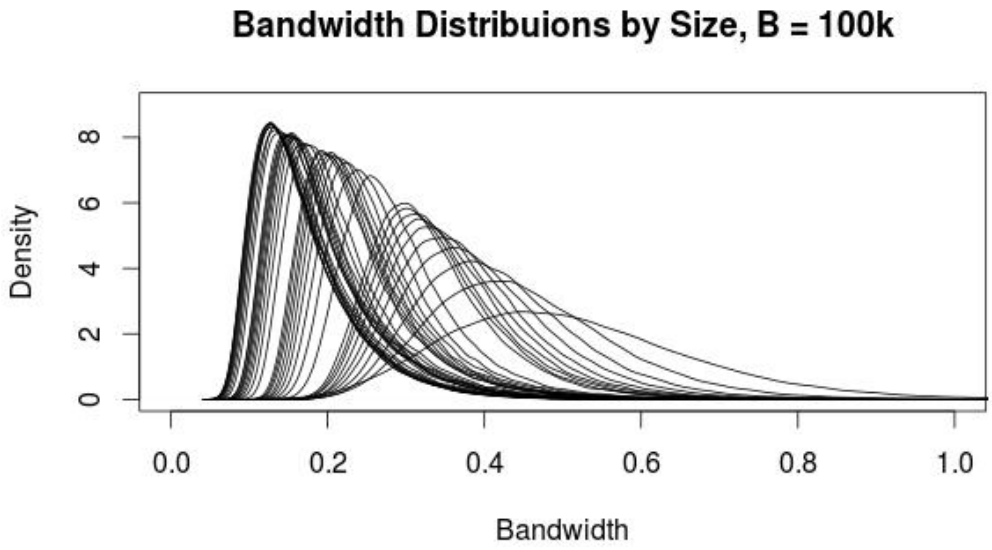

Figure 17: Distributions of bandwidths as sample size decreases (left to right)

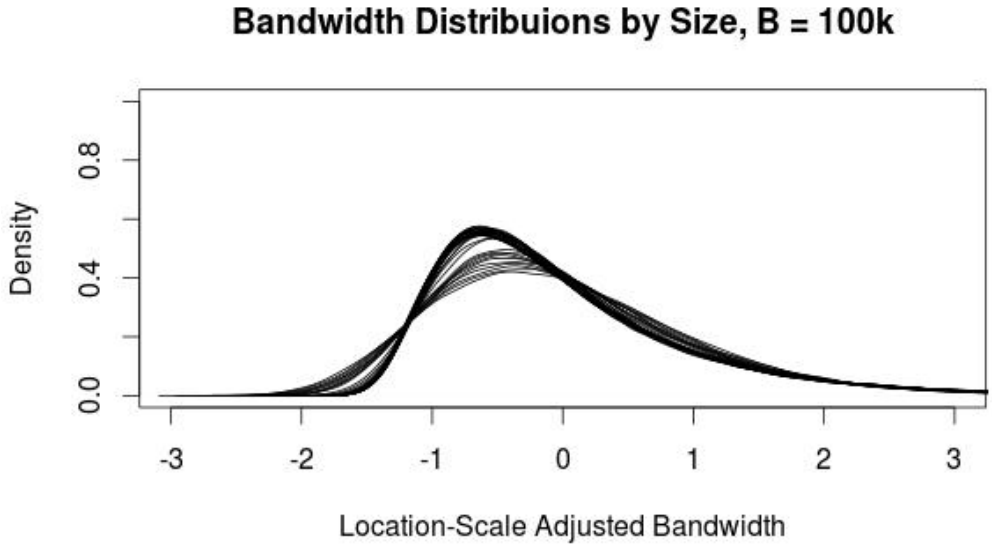

Figure 18: Location-scale adjusted distributions of bandwidths by sample size.

It appears that there is a relationship between data size and the resulting distribution, just as Jones (1983) suggests based on his estimated proportions of multimodal results using calculated bandwidths. The relationship is best illustrated by sorting the results and calculating the bandwidth in each of the distributions that represents the 5% cut-off point for comparison (Figure 19).

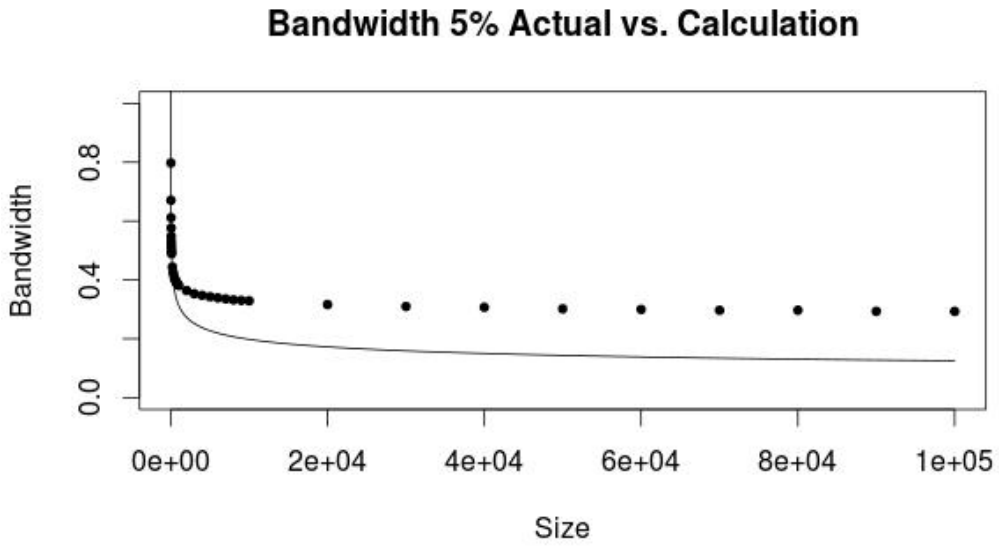

Figure 19: Bandwidth at upper 5% of each distribution by size (dots) and $h = 1.25\sigma n^{-1/5}$ *(line).*

Jones's (1983) formula in (Figure 19) shows $h = 1.25\sigma n^{-1/5}$ compared to the results found obtained from the samples, showing that the Jones formula is close and why it works for smaller data sizes but fails as data sizes increase. The same relationship holds for the mean and standard deviation of each of the bandwidth distributions (Figures 19 & 20).

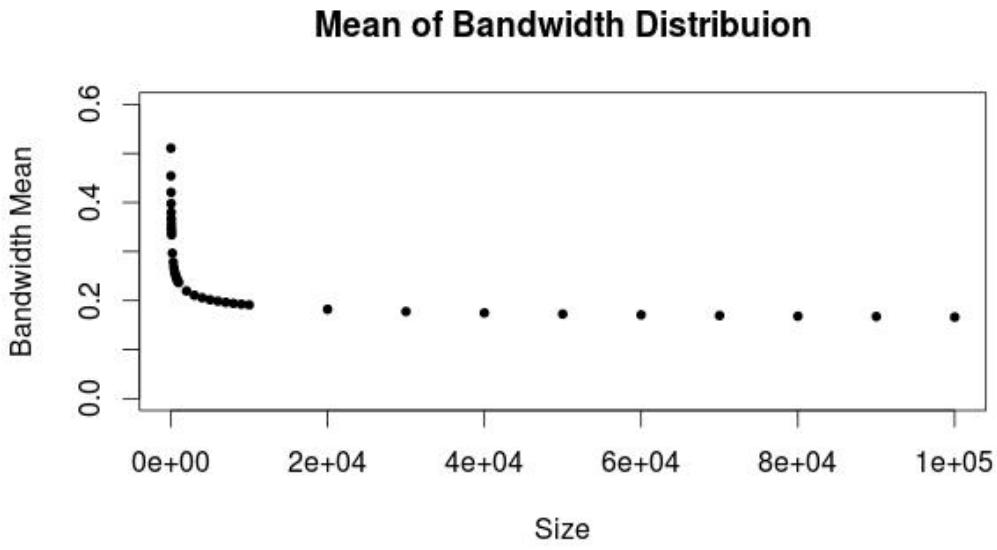

Figure 20: Mean of each distribution of bandwidths by size.

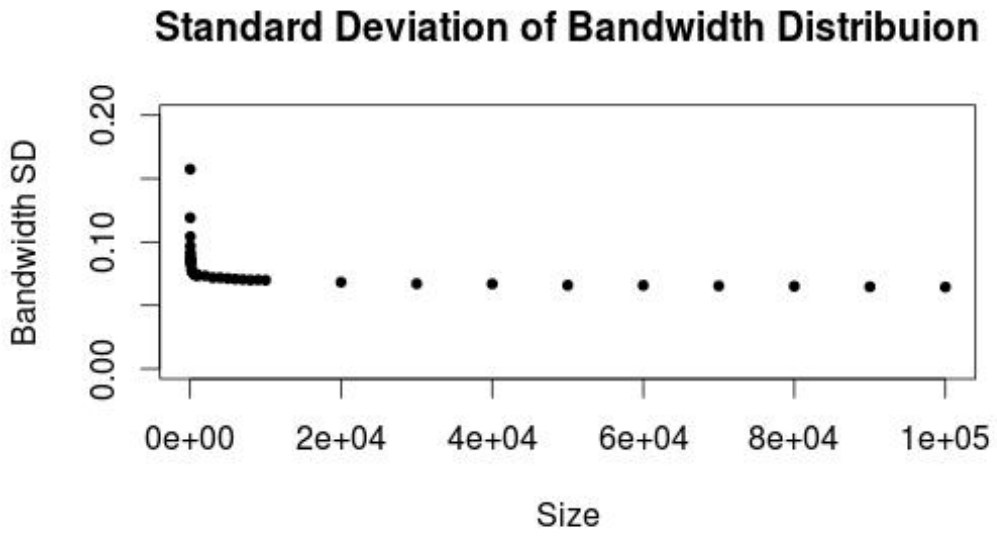

Figure 21: Standard deviation of each distribution of bandwidths by size.

Combining this relationship with a location-scale adjustment on the distribution of bandwidths, the critical bandwidth found for the data can now be compared to a "standardized" result, which can be calculated. To find this result, find the upper 5% bandwidth point for each distribution of bandwidths. Next, perform a location-scale adjustment to see if the 5% cut-off points appear to converge at an identifiable point (Figure 22).

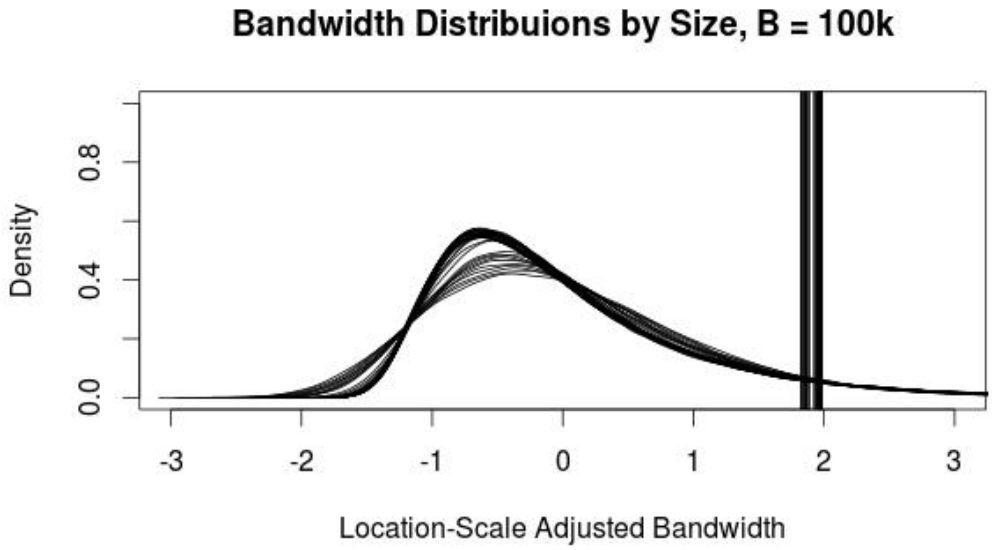

Figure 22: Location-scale adjusted distributions of bandwidths with p-value of 5% (solid lines).

This result shows that the 5% boundary occurs around 2 standard deviations after location-scale adjusting the distributions of bandwidths, which is consistent with observations on mixtures of normal components (Cox, 1966). The suggested adaptations to the modality tests

appear to work well for unimodal conditions, with the next logical extension determining the best mode for a dataset.

3.5 Determining Modality Beyond Unimodal

A test for unimodality versus multimodality is a good start, but testing for the presence of other modes is also desirable. There are some fields of study where the interest is in data that has two or three modes, and unimodality testing is unable to provide a thorough answer. The addition of a few new measurements makes it possible to extend the modified test for unimodality to test for virtually any modality. Because the kernel density function in use is the normal density function, each mode in a multimodal case will follow a normal distribution, but with different parameters (Figure 23). These additional parameters are “distance,” a measurement of the mean of the mode relative to the mean of the data, and “weight,” a measurement of the proportion of data within the mode.

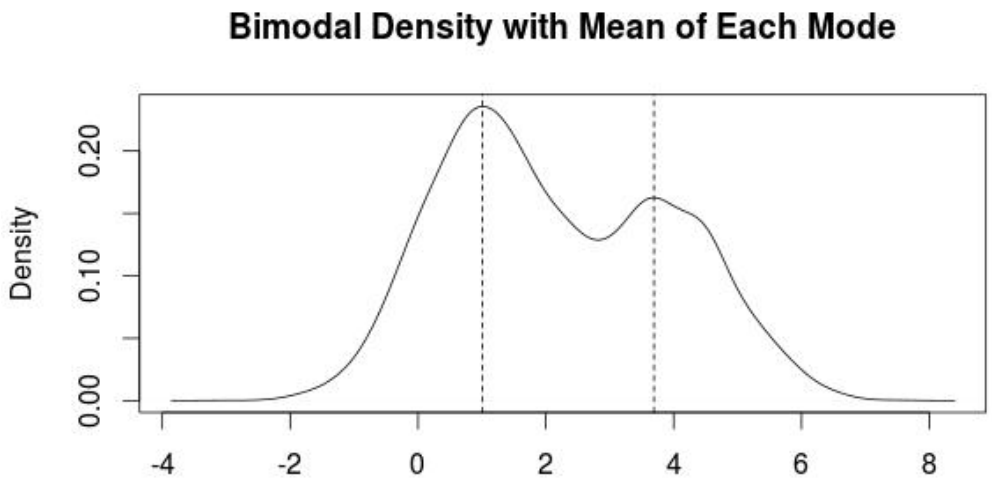

Figure 23: Example of bimodal density with mean of each mode (dashed line).

As with the unimodal test, a distribution function for the chosen mode is constructed using parameters obtained from the dataset for each mode of interest. Samples are drawn from the constructed distribution function, and the extreme bandwidth correlating to the desired mode is determined. A comparison between the extreme bandwidth, found for the data at the desired mode, and the distribution of bandwidths, obtained from the samples, can now be conducted (Figure 24).

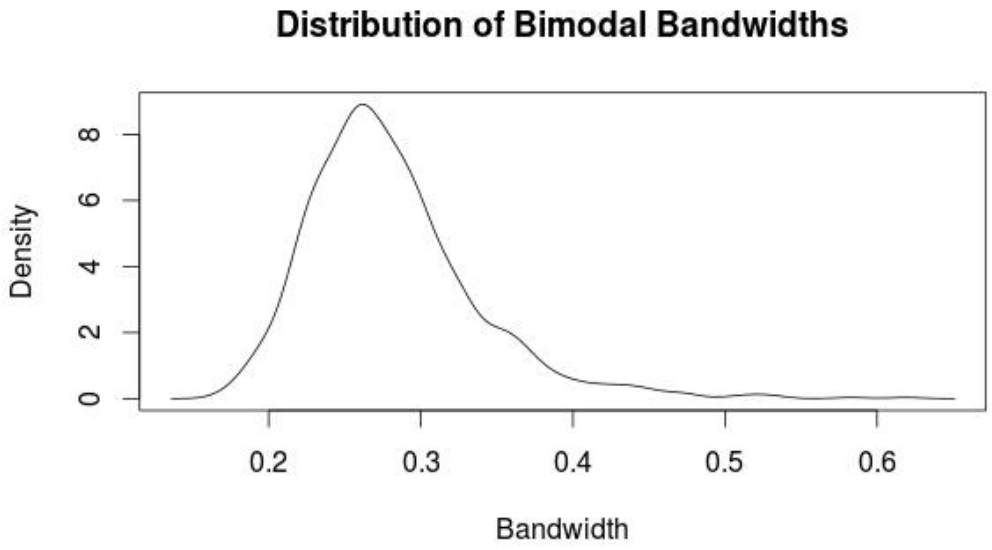

Figure 24: Distribution of extreme bimodal bandwidths found in samples of bimodal density.

The distribution of bandwidths generated for the desired mode follows a similar distribution found in the unimodal testing. In fact, when these two distributions are location-scale adjusted, the distributions are nearly identical (Figure 25). This is an indication that there is a measurable relationship between all of the parameters at any given mode.

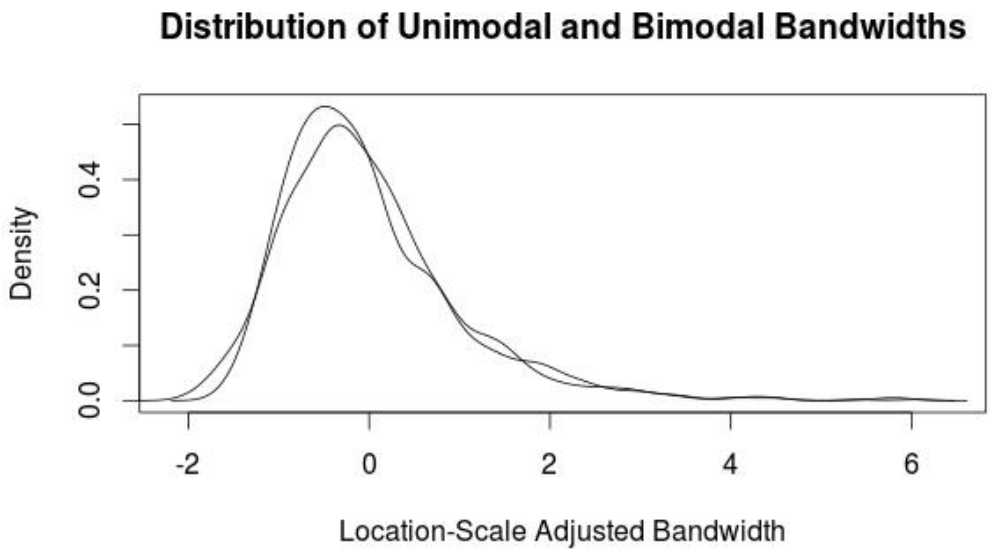

Figure 25: Location-scaled adjusted bandwidth results from unimodal and bimodal sampling.

The relationship holds for multiple modes tested, but the complexity of the interaction of variables increases. Number of modes, distance of means, and weight of data points will all affect the constructed density function. Location-scale adjustments, in theory, continue follow a similar relationship for any bandwidth distribution function.

It is important to construct the modal density function so it accurately represents each mode. Initially, this is accomplished using the R routine 'normalmixEM' found in the package 'mixtools'. However, as the number of modes increases, this routine returns mixed distribution results that do not match with the desired distribution. Because the parameters of each mode are known, weighted samples can be drawn from the normal density function representing each mode using an R routine to find the modes and generate samples (Appendix B.8 and B.9). The

extreme bandwidth found for the desired mode can then be compared against the distribution of bandwidths found in the samples.

3.6 Application of New Methodology

The meteor dataset, plotted to a linear reference, appears to contain three modes visually (Figure 26). The dataset contains 22 observations; each observation is unique at two decimal places. This means the maximum possible number of modes is 22, but it is only expected if the data has sufficient space between each observation. The clustering of the data should substantially reduce the number of modes found.

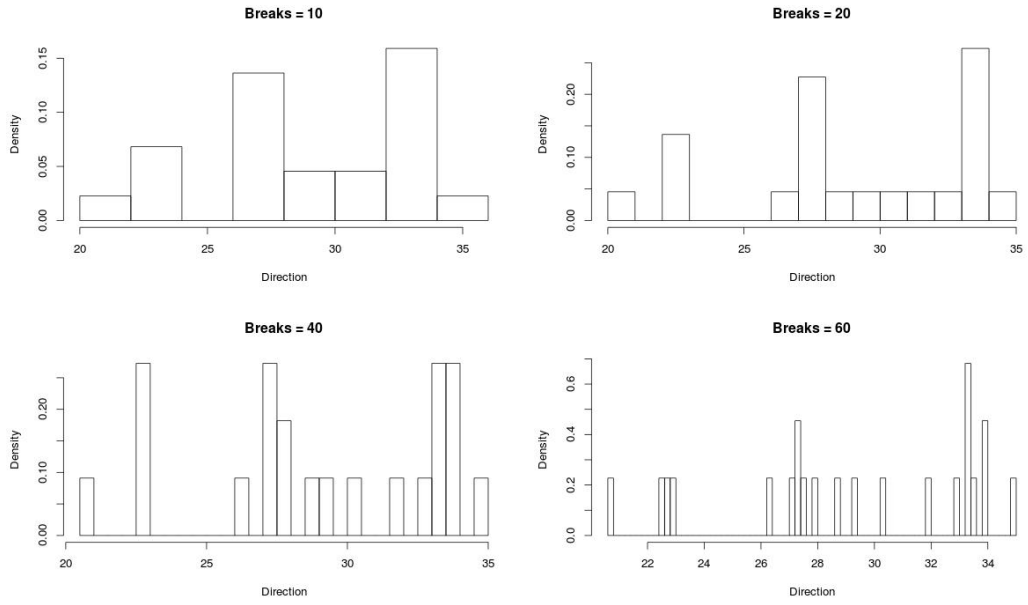

Figure 26: Histograms of meteor data at different breaks to visualize modality.

Varying the number of breaks used in the histogram changes the visual perspective of the dataset. Fewer breaks make the data appear unimodal and more breaks make the data appear trimodal. This is still simply guessing at the modality based upon appearance and each mode should be tested for statistical significance utilizing the proposed methodology (Figure 27).

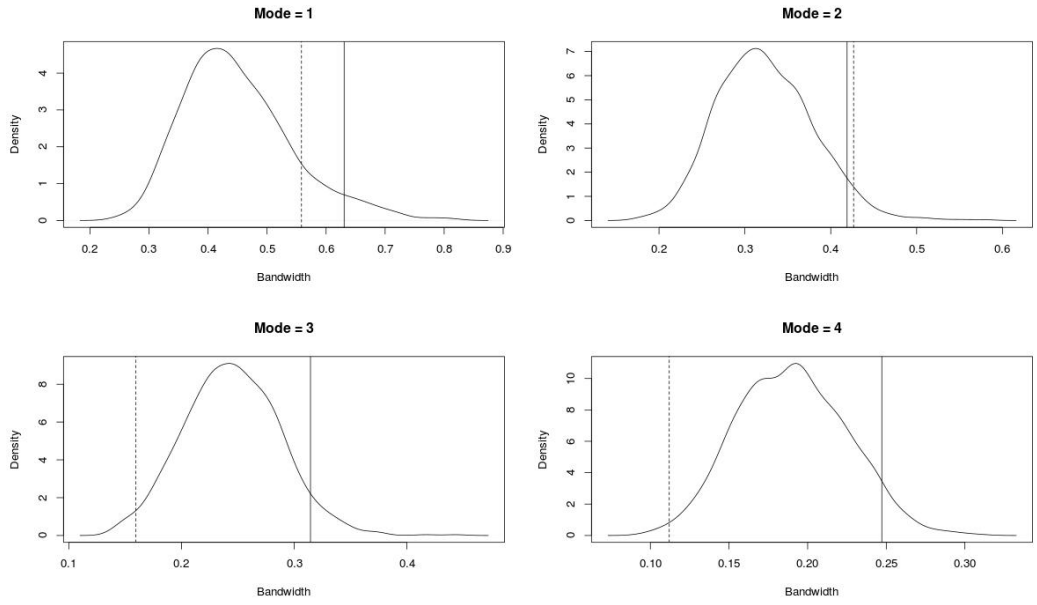

Figure 27: Results of modified test for modality showing distribution of bandwidths, p-value of 5% (solid line), and critical bandwidth found at each mode (dashed line).

Based on tests for statistical significance, modes 1, 3, and 4 are plausible; however, mode 2 is not. While a mode of 4 is possible, oversmoothing should be considered at this point because it causes minor clusters to be counted as modes, which can also be seen in the histograms. This

leaves unimodal and trimodal as prospective candidates to best estimate the modality of the dataset.

The azimuth dataset contains many more data points then the meteor dataset, and the histogram indicates a number of modes, possibly eight or more (Figure 28). It is much harder to determine the modality of this dataset visually due to the size of the dataset. Testing for modality is the only way to determine credible modes in this case (Figures 29 & 30).

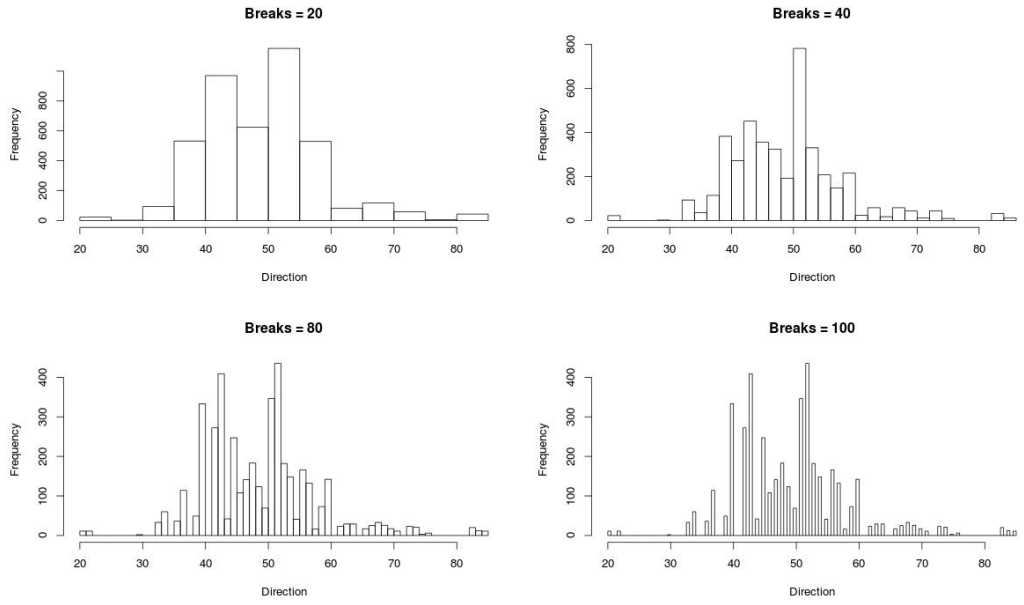

Figure 28: Histograms of azimuth data at different breaks to visualize modality.

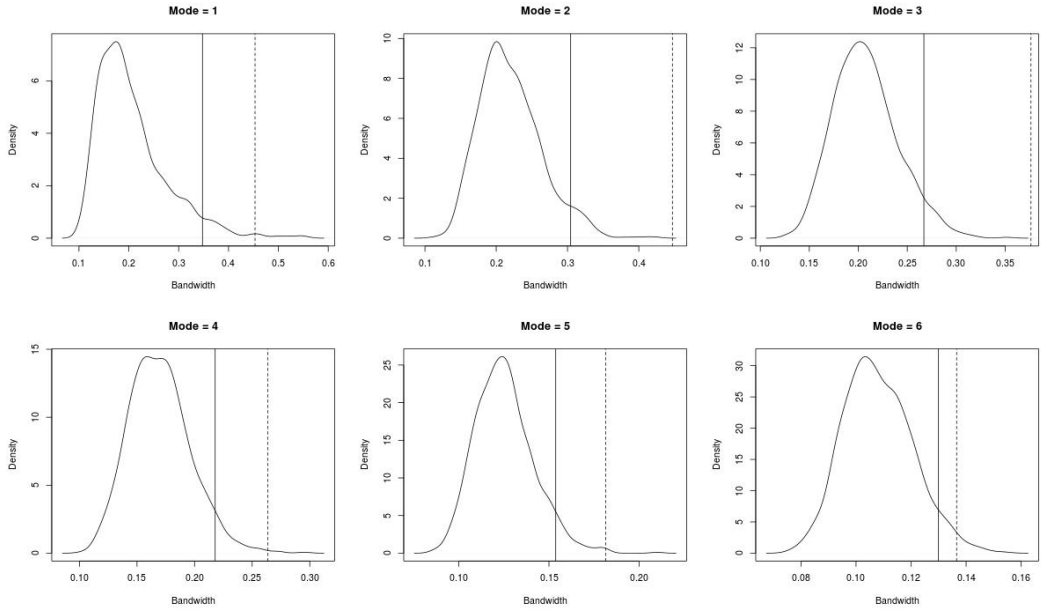

Figure 29: Results of modified test for modality showing distribution of bandwidths, p-value of 5% (solid line), and critical bandwidth found at each mode (dashed line).

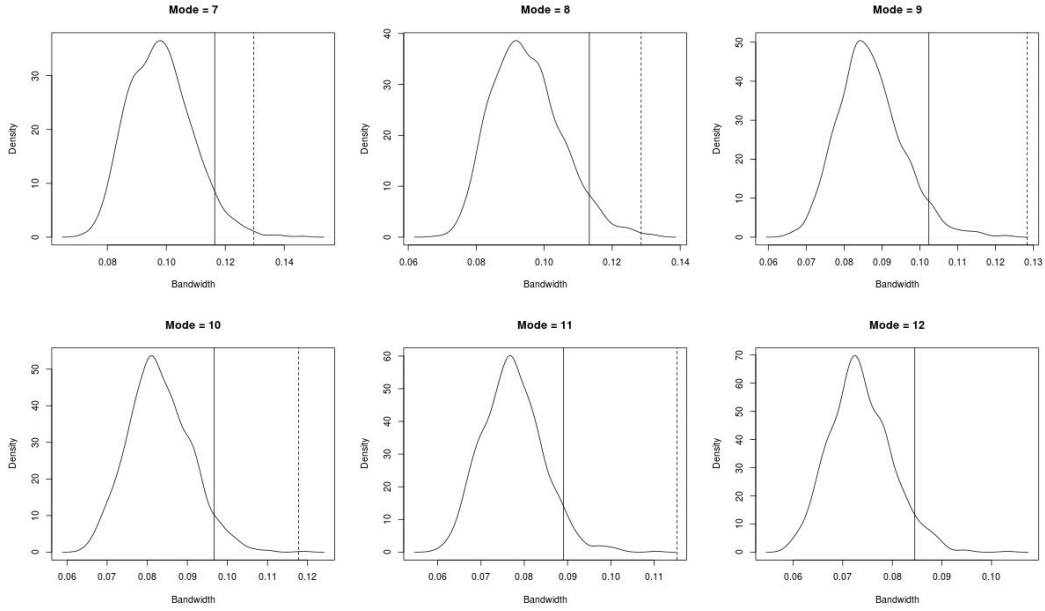

Figure 30: Continued results for further modes.

The results show that there are no statistically significant modes as tested through 12 modes. A mode of 6 may be considered due to the fact it is closest to the p-value, but still not statistically significant. In addition, it is possible that a more likely candidate for mode exists beyond 12. These are problematic results because there are so many clusters of data it is hard to determine the best mode. While testing is possible for virtually any mode, the desire may not be to find the exact mode, but possible modes within a range or major modes within the data. Such is the case with the azimuth data to determine if it contains 1, 2 or 3 modes. Perhaps removing the data that makes up the minor nodes, which affects the ability to detect dominant modes, and retesting will yield better results.

The azimuth dataset is trimmed and re-tested with observations below 30 and above 64 removed from the dataset (Figures 31 & 32). The removed observations created several small outlying modes that detract from the main cluster of data. The data is still markedly multimodal and the possibility of bimodal is now visually evident.

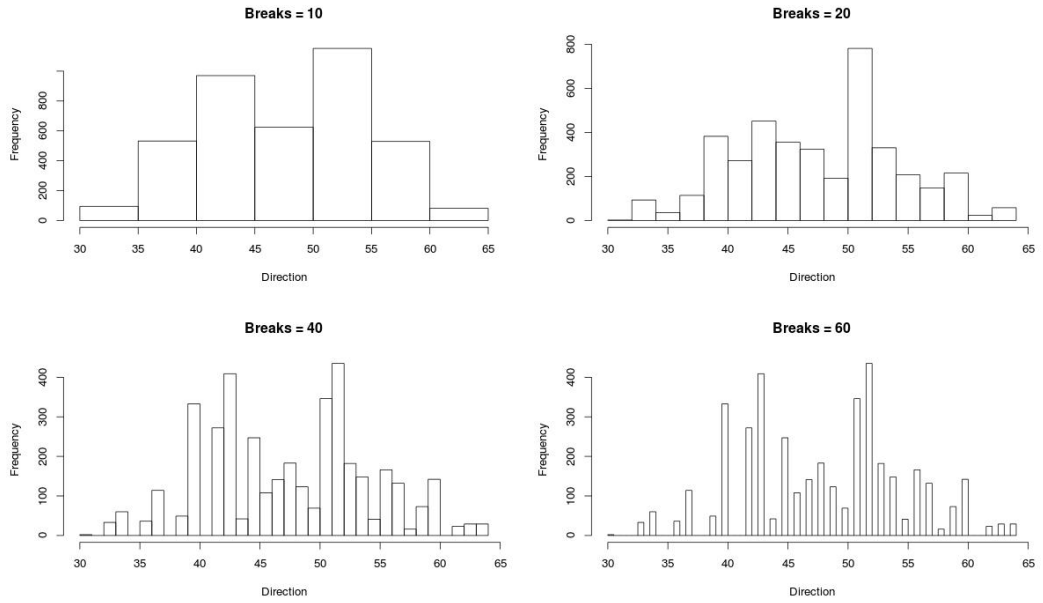

Figure 31: Histograms of trimmed azimuth data at various breaks to visualize modality.

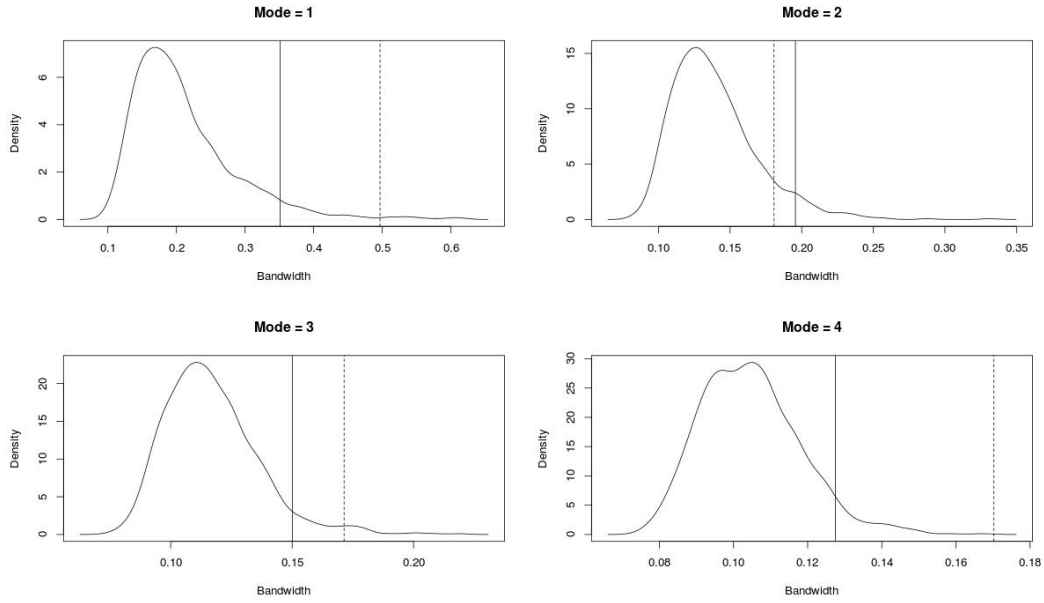

Figure 32: Results of modified bandwidth tests showing distribution of bandwidths, p-value of 5% (solid line), and critical bandwidth for each mode (dashed line).

Testing of the first four modes now indicates a modality of 2 is credible. While the dataset has a high actual modality, there are two modes where a majority of the data appears to be clustered. Based on the results, calculations to determine how fluid disseminates through the network of fractures could be considered bimodal, keeping in mind that the results are from linear testing rather than circular testing and the data was trimmed. The statistical tests for modality appear to be successful.

CHAPTER 4

CONCLUSION

The tests for unimodality, proposed by Silverman (1986), work but only for datasets with a limited number of observations and are restricted to unimodal testing only. Silverman's (1986) approach draws samples from a nearly bimodal distribution function created from the data. This results in more samples becoming bimodal as the number of observations increases because large data sizes represent the distribution more closely, thus it is easier to obtain a multimodal result. Sampling instead from a known unimodal distribution, such as the normal distribution with parameters matching that of the dataset, will eliminate the tendency towards multimodal results as the number of observations increases. Similarly, drawing samples from a mixed distribution composed of multiple known unimodal distributions allows the natural extension to test for virtually any modality.

Modified statistical test for modality:

1. Given a dataset $X_1, \ldots, X_n$, calculate mean = μ and standard deviation = σ.
2. Location-scale adjust the dataset so that mean = 0 and standard deviation = 1.
3. Find h_m, the extreme critical bandwidth for mode m.
4. Find f_m, the extreme density function with bandwidth h_m.
5. Find the location of each mode, $d_1, \ldots, d_m$, distance from mean = 0.

6. Find the weight of each mode, $w_1, \ldots, w_m$, proportion of data in mode m.
7. Generate a sample for each mode m with size = $n*w_m$, mean = d_m, and standard deviation = 1, such that the total sample size is n.
8. Find the extreme critical bandwidth for mode m of the sample.
9. Repeat (7) and (8) a large number of times, B.
10. Find h_p, the bandwidth at desired p-value.
11. If $h_m < h_p$, then m is a credible mode for the data.
12. Repeat (3) to (11) for each mode m.

Simulation studies conducted by Jones (1983) indicate that bandwidth and modality are linked to a function, but again only for datasets with a limited number of observations and restricted to unimodal functions. Further simulation studies conducted on datasets, including those of larger sizes, show that bandwidth measurements indeed have certain properties. The most important property is the distribution of critical bandwidths for any chosen mode follows a gammalike distribution, which shifts based on the parameters of the density function. Because the distributions of bandwidths are all related, the mean and standard deviation of each distribution both follow an exponential distribution. These two properties together should allow a quick calculation to be performed on the critical bandwidth found for the data and then a comparison made to a "standardized" gamma function. The "standardized" gamma function is a location-scale adjusted version of any distribution of bandwidths obtained through simulation. This removes the need for computationally intensive bandwidth calculations.

Simplified statistical test for modality:

1. Given a dataset $X_1, \ldots, X_n$, calculate mean = μ and standard deviation = σ.
2. Location-scale adjust the dataset so that mean = 0 and standard deviation = 1.
3. Find h_m, the extreme critical bandwidth for mode m.
4. Calculate μ_m and σ_m, based on exponential relationship.
5. Location-scale adjust h_m using $z_m = (h_m - \mu_m) / \sigma_m$.
6. Find z_p, the "standardized" bandwidth at desired p-value, which may vary slightly based on range of data size.
7. If $z_m < z_p$, then m is a credible mode for the data.
8. Repeat (1) to (8) for each mode m.

Note that tests for modality only provide plausible candidates for mode selection. It is necessary to have a good understanding of the data, why specific modes are important, and to visualize the data using histograms. Datasets with a small number of observations can be made to fit nearly any mode, whereas those with a large number of observations may be hard to fit lower modalities.

Simulation studies provide evidence showing a single function may make it possible to calculate a "standardized" critical bandwidth, which can then be compared to a p-value from the "standardized" distribution of bandwidths. However, the exact distribution of the means and standard deviations of the "standardized" distribution of bandwidths based on data size, location and weight of mode needs to be determined more accurately. Additionally, there still appears to be issues with the number of observations and the mode results. Further studies are necessary to

understand how to mitigate the impact of the data size on mode. While the simulation studies indicate statistical testing for modality is possible, mathematical proofs are still necessary to provide further justification to this theory. Last, while this method should hold when performed in a circular frame of reference, further research on application to circular data and functions is essential because properties and results may differ.

REFERENCES

Cox, D. R. (1966). Notes on the analysis of mixed frequency distributions. *British Journal of Math and Statistical Psychology, 19*, 39-47.

Efron, B. (1981). Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. *Biometrika, 68(3)*, 589-599.

Ekneligoda, T. C., & Henkel, H. (2010). Interactive spatial analysis of lineaments. *Journal of Computers and Geosciences, 36*, 1081-1090.

Fisher, N. I. (1993). *Statistical analysis of circular data*. Cambridge, England: Cambridge University Press.

Good, I. J., & Gaskins, R. A. (1980). Density estimation and bump-hunting by the penalized likelihood method exemplified by scattering and meteorite data. *Journal of the American Statistical Association, 75*, 42-56.

Jones, M. C. (1983). The projection pursuit algorithm for exploratory data analysis (Doctoral dissertation, University of Bath, 1983).

Manzocchi, T. (2002). The connectivity of two-dimensional networks of spatially correlated fractures. *Journal of Water Resources Research, 38-9*, 1162-1181.

Silverman, B. W. (1986). Chapter 6: Density estimation in action. In *Density estimation for statistics and data analysis* (pp. 137-147). New York, NY: Chapman and Hall.

Taylor, C. C. (2008). Automatic bandwidth selection for circular density estimation. *Computational Statistics & Data Analysis, 52*(7), 3493 -3500.

APPENDIX A

DATASETS

A.1 FisherB13c, Set 1, Termite Mound Orientations, Circular

structure(c(161, 182, 179, 193, 164, 166, 144, 175, 163, 187, 177, 161, 170, 169, 144, 179, 175, 185, 211, 176, 184, 149, 166, 173, 144, 174, 202, 170, 164, 160, 163, 218, 181, 161, 180, 218, 18, 202, 152, 140, 244, 187, 203, 187, 180, 230, 190, 200, 194, 181, 192, 168, 164, 171, 179, 166, 174, 164, 166, 257, 215, 208, 187, 212, 177, 186, 171, 196, 188, 188, 163, 201, 204, 184, 218, 220, 178, 316, 161, 182, 180, 200, 211, 228, 168, 197, 202, 273, 158, 150, 157, 182, 189, 174, 136, 202, 202, 167, 181, 193), circularp = structure(list(type = "angles", units = "degrees", template = "none", modulo = "asis", zero = 0, rotation = "counter"), .Names = c("type", "units", "template", "modulo", "zero", "rotation")), class = c("circular", "numeric"))

A.2 FisherB13, Set 1, Termite Mound Orientations, Linear

c(161, 182, 179, 193, 164, 166, 144, 175, 163, 187, 177, 161, 170, 169, 144, 179, 175, 185, 211, 176, 184, 149, 166, 173, 144, 174, 202, 170, 164, 160, 163, 218, 181, 161, 180, 218, 18, 202, 152, 140, 244, 187, 203, 187, 180, 230, 190, 200, 194, 181, 192, 168, 164, 171, 179, 166, 174, 164, 166, 257, 215, 208, 187, 212, 177, 186, 171, 196, 188, 188, 163, 201, 204, 184, 218, 220, 178, 316, 161, 182, 180, 200, 211, 228, 168, 197, 202, 273, 158, 150, 157, 182, 189, 174, 136, 202, 202, 167, 181, 193)

A.3 Cracks.A, Azimuth Orientation of Cracks, Circular

structure(c(20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 30, 30, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 37, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 40,

40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
40, 40, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42,
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,

43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,

43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,

43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,

43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,

43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,

43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,

43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,

43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,

43, 43, 43, 43, 43, 43, 43, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44,

44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45,

45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46,

46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46,

46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46,

46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46,

46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47,

47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47,

47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47,

47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47,

47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47,

47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 47, 48,

48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48,

48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48,

48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48,

48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48,

48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48,

48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48,

48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48,

49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49,

49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49,

49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49,

49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49,

49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 50, 50, 50, 50, 50, 50, 50,

50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50,

50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50,

50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,

51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51,
51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 52, 52, 52, 52, 52, 52, 52, 52,
52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52,
52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52,
52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52,
52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52,
52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52,
52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52,
52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52,
52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52,
52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52,

52, 53, 54, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 56, 56, 56, 56, 56, 56, 56, 56,

56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56,

56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56,

56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56,

56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56,

56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56,

56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56,

56, 56, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57,

57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57,

57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57,

57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57,

57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57,

57, 57, 57, 57, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 59, 59, 59, 59, 59, 59,

59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59,

59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59,

59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,

60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,

60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,

60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,

60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,

60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,

60, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 63, 63,

63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63, 63,

63, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 68, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 69, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71, 71, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 73, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 74, 75, 75, 75, 76, 76, 76, 76, 76, 76, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 83, 84, 84, 84, 84, 84, 84, 84, 84, 84, 84, 84, 84, 85, 85, 85, 85, 85, 85, 85, 85, 85, 85, 85), circularp = structure(list(type = "angles", units = "degrees", template = "none", modulo = "asis", zero = 0, rotation = "counter"), .Names = c("type", "units", "template", "modulo", "zero", "rotation")), class = c("circular", "numeric"))

APPENDIX B

R CODE, ROUTINES, FUNCTIONS

B.1 Find Critical Bandwidth for Circular Data

```
function(data, mc.test=1, precision=6, plot=FALSE) {
        # mc.test is the mode count to test
        # i.e. if test for unimodal, then mc.test = 1
        # i.e. if test for bimodal, then mc.test = 2, etc.

        require(circular)

        # bandwidth increases rapidly for circular data so check integers first
        max.exp = 5
        # lower this if there are errors calculating mode count
        mci = 100
        # fast check by factor of 10
        while (mci > mc.test ) {
                h.integer = 10^max.exp
                mci <- circ.mode.count(density(data, bw = h.integer, control.circular = list(units =
                "degrees")))
                max.exp = max.exp - 1
        }
        # h.integer is now below the bandwidth we seek
        # incremental check
```

```
max.exp = max.exp + 1
h.integer = h.integer + 10^max.exp
for (i in max.exp:0) {
        while (mci < mc.test + 1 ) {
                h.integer = h.integer + 10^(i)
                mci <- circ.mode.count(density(data, bw = h.integer, control.circular =
                list(units = "degrees")))
        }
        h.integer = h.integer - 10^(i)
        mci <- circ.mode.count(density(data, bw = h.integer, control.circular = list(units =
        "degrees")))
}

# initial h.upper must be greater than zero
# so add one to precision
h.upper = h.integer + 10^-(precision+1)
# set to a mode count that is not likely
mc = 0

for (n in 1:precision) {
        while (mc < mc.test + 1) {
                h.upper = h.upper + 10^-n
```

```
                mc <- circ.mode.count(density(data, bw = h.upper, control.circular =
                list(units = "degrees")))
            }
            h.upper = h.upper - 10^-n
            mc <- circ.mode.count(density(data, bw = h.upper, control.circular = list(units =
            "degrees")))
        }
        # adjust h.upper to account for added level of
        # precision that was initially set
        # otherwise density will yield the wrong number of
        # of modes, recall that h.upper is intended to
        # represent the extreme bandwidth that will still
        # yield the desired mode count
        h.upper = h.upper + 2*10^-precision - 10^-(precision+1)

        if (plot == TRUE) {
            plot(circular(data, units="degrees"), stack=TRUE, bins=150, shrink=2.0)
             lines(density(data, bw = h.upper, control.circular = list(units = "degrees")), col =
            'blue')
        }
        return(h.upper)
}
```

B.2 Generate Samples from Circular Density and Calculate Multimodal Proportion

```
function(data,dmode=1,dbw=30,dsamp=100,timer=TRUE) {
	# data = dataset name
	# mode = mode limit to test, default 1
	# bw = bandwidth to use for density
	# samples = number of samples to take, default = 100

	require(circular)
	require(parallel)
	require(foreach)
	require(doParallel)

	corecount <- detectCores()
	usecores = 1
	if (corecount > 2) {
		usecores = corecount - 2
	}
	registerDoParallel(cores=usecores)

	#cat('Found',corecount,'cores : using',usecores,'for computations.\n')
```

```
fhat <- density(data, bw=dbw, control.circular=list(units="degrees"))
nextmode <- 0
runtime <- system.time(
        outcome <- foreach(i=1:dsamp, .combine=c) %dopar% {
                nextmode <- 0
                testit <- sample(fhat$data, length(data), replace=TRUE, prob=NULL)
                mc <- circ.mode.count(density(testit, bw = dbw, control.circular =
                list(units = "degrees")))
                if (mc > dmode) {
                        nextmode <- 1
                }
                nextmode
        }
)

if (timer == TRUE) {
        #cat('-----------------------------------------------\n')
        #cat('Elapsed runtime : ', format(.POSIXct(runtime[3],tz = "GMT"),
        "%H:%M:%S"), '\n')
        #cat('-----------------------------------------------\n')
}
```

```
        result <- 0
        for (j in 1:dsamp) {
                result <- result + outcome[j]
        }

        proportion <- result / dsamp
        return(proportion)
}
```

B.3 Count the Number of Modes in Circular Sample Density

```
function(ckde) {
        mc <- 0
        for(i in 2:(length(ckde$x)-1))
        if((ckde$y[i-1]<ckde$y[i])&&(ckde$y[i]>ckde$y[i+1])) mc <- mc + 1
        return(mc)
}
```

B.4 Find Critical Bandwidth for Linear Data

```
function(data, mc.test=1, precision=6) {
        # mc.test is the mode count to test
        # i.e. if test for unimodal, then mc.test = 1
        # i.e. if test for bimodal, then mc.test = 2, etc.

        # initial h.upper must be greater than zero
        # so add one to precision
        h.upper = 10^-(precision+1)
        # set to a mode count that is not likely
        mc = 100

        for (n in 1:precision) {
                while (mc > mc.test) {
                        h.upper = h.upper + 10^-n
                        mc <- mode.count(density(data,h.upper))
                }
                h.upper = h.upper - 10^-n
                mc <- mode.count(density(data,h.upper))
        }
        # adjust h.upper to account for added level of
```

```
        # precision that was initially set
        # otherwise density will yield the wrong number of
        # of modes, recall that h.upper is intended to
        # represent the extreme bandwidth that will still
        # yield the desired mode count
        h.upper = h.upper + 2*10^-precision - 10^-(precision+1)

        return(h.upper)
}
```

B.5 Generate Samples from Linear Density and Calculate Multimodal Proportion

```
function(data=chond,mc=1,samples=10000) {
        # function to test the distributions of h.crit values

        # setup for parallel processing
        require(parallel)
        require(foreach)
        require(doParallel)

        corecount <- detectCores()
        usecores = 1
```

```
if (corecount > 2) {
        usecores = corecount - 2
}
registerDoParallel(cores=usecores)

#data <- Cracks.A.ca
data.size <- 10 # c(20, 40 ,60 ,80 ,200 ,400 ,600 ,800 ,2000 ,4000 ,6000 ,8000 ,20000
,40000 ,60000 ,80000)
data.sd <- 1.0 # sd(data) # c(0.05, 1.0, 2.5) # c(0.01, 0.05, 0.10, 0.50, 1.0, 1.5, 2.0, 2.5,
3.0)
data.mean = 0 # mean(data)
#data.gap = 10

results <- array(dim=c(length(data.size),2,samples))
# generate unimodal distribution(s) using rnorm
#tt <- c()
#for (t in 1:1) {
#    runtime <- system.time(
        for (i in 1:length(data.size)) {
            #for (j in 1:length(data.sd)) {
                bws <- foreach(k=1:samples, .combine=c) %dopar% {
```

```
                    gen <- rnorm(data.size, data.mean, data.sd) # rnorm(data.size[i],
                    data.mean, data.sd[j])
                    # gen <- mode.sample(data,mc) # for multi modal tests
                    fhc(gen,mc)
                }
                results[i,1,] <- bws
                ls.bws <- (bws - mean(bws))/sd(bws)
                results[i,2,] <- ls.bws
            #}
        }
    #   )
    #   tt <- c(tt, runtime[3])
    #}
    #avg.tt <- sum(tt)/length(tt)
    #cat(avg.tt,"\n")
    return(results)
}
```

B.6 Count the Number of Modes in Linear Sample Density

```
function(kde) {
    mc <- 0
```

```
        for(i in 2:(length(kde$x)-1))
        if((kde$y[i-1]<kde$y[i])&&(kde$y[i]>kde$y[i+1])) mc <- mc + 1
        return(mc)
}
```

B.7 Test an Array of Different Data Sizes and Standard Deviations

```
function(useparallel = 1) {
        # set size, mean, sd and perform the same test numerous times
        # to make sure that the precision is consistant

        if (useparallel) {
                # setup for parallel processing
                 require(parallel)
                require(foreach)
                require(doParallel)

                corecount <- detectCores()
                usecores = 1
                if (corecount > 2) {
                        usecores = corecount - 2
                }
```

```
        registerDoParallel(cores=usecores)


        cat('Found',corecount,'cores : using',usecores,'for computations.\n')
}


# unimodal


# set initial values
size = c(50, 100, 500, 1000, 5000, 10000) # claimed range 40 – 5000
        # c(50, 100, 500, 1000, 5000) for later tests
mean = 0
sd = c(0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0)
        # c(0.01, 0.05, 0.1, 0.5, 1.0, 2.0, 3.0) for later tests
mc.test = 1
samples = 1000 # this is precision


td <- function(x, k=nchar(samples)-1) {
        format(round(x, k), nsmall=k)
}


for (i in sd) {
        cat("   ",td(i))
```

```
        }
        cat("\n")

        for (i in size) {
                for (j in sd) {
                        result <- foreach(k=1:samples, .combine='+') %dopar% {
                                test.jones83(i, mean, j, mc.test)
                        }
                        cat("   ",td(result/samples))
                }
                cat("\n")
        }
}
```

B.8 Find Modes and Calculate Parameters

```
function(data=chond,mc=1,error=0.05,plots=TRUE) {
        # While smaller values for error will generate a density function
        # with parameters close to that of the original data
        # they will also take longer to generate, so take care in
        # how small error is set
        # This functions will simulate a multimodal distribution
```

```
# using parameters of the orginal data
# the simulated data can be used to test what percent of
# the time the calculated mode is beyond the actual mode
# Then, compare the proportions of data and sim
data.hc <- fhc(data,mc)
data.df <- density(data,data.hc)
xpoints <- mode.find(data.df)
ypoints <- c()
for (i in xpoints) {
        ypoints <- c(ypoints, density(data,data.hc,from=i,to=i,n=1)$y)
}

data.mean <- mean(data)
data.sd <- sd(data)
# set the acceptable margin of error allowed
# warning, setting this to low will increase loop time
# possibly causing an infinite loop
error.sd <- data.sd * error
error.bw <- data.hc * error
# set the initial errors higher to start the loop
diff.sd <- error.sd * 2
diff.bw <- error.bw * 2
```

```
# keep simulating datasets until parameters are within margin of error
keep.going <- TRUE
while (keep.going) {
        sim <- c()
        for (j in 1:mc) {
                sim.wt <- ypoints[j] / sum(ypoints)
                sim.size <- round(length(data) * sim.wt)
                sim.asd <- data.sd
                sim <- c(sim,rnorm(n=sim.size,mean=xpoints[j],sd=sim.asd))
         }
        sim.sd <- sd(sim)
        sim.hc <- fhc(sim,mc)
        diff.sd <- abs(sim.sd-data.sd)
        diff.bw <- abs(sim.hc-data.hc)
        if((diff.sd<=error.sd)&&(diff.bw<=error.bw)) {
                keep.going <-FALSE
        }
        #cat('OUT',(diff.sd>error.sd),(diff.bw>error.bw),'\n')
}
sim.df <- density(sim,sim.hc)
sim.mean <- mean(sim)
```

```
        if(plots) {
                #par(mfrow=c(1,2))
                 plot(data.df,pch='.',main='Data')
                abline(v=data.mean,col='blue')
                abline(v=xpoints,col='red')
                lines(sim.df,pch='.',main='Sim',col='red')
                #abline(v=data.mean,col='blue')
                #abline(v=xpoints,col='red')
                #par(mfrow=c(1,1))
        }

        return(sim)
}
```

B.9 Create Data from Parameters of Found Modes

```
function(data=chond,mc=1,error=0.05,plots=TRUE) {
        # While smaller values for error will generate a density function
        # with parameters close to that of the original data
        # they will also take longer to generate, so take care in
        # how small error is set
        # This functions will simulate a multimodal distribution
```

```
# using parameters of the orginal data
# the simulated data can be used to test what percent of
# the time the calculated mode is beyond the actual mode
# Then, compare the proportions of data and sim
data.hc <- fhc(data,mc)
data.df <- density(data,data.hc)
xpoints <- mode.find(data.df)
ypoints <- c()
for (i in xpoints) {
        ypoints <- c(ypoints, density(data,data.hc,from=i,to=i,n=1)$y)
}

data.mean <- mean(data)
data.sd <- sd(data)
# set the acceptable margin of error allowed
# warning, setting this to low will increase loop time
# possibly causing an infinite loop
error.sd <- data.sd * error
error.bw <- data.hc * error
# set the initial errors higher to start the loop
diff.sd <- error.sd * 2
diff.bw <- error.bw * 2
```

```
# keep simulating datasets until parameters are within margin of error
keep.going <- TRUE
while (keep.going) {
	sim <- c()
	for (j in 1:mc) {
		sim.wt <- ypoints[j] / sum(ypoints)
		sim.size <- round(length(data) * sim.wt)
		sim.asd <- data.sd
		sim <- c(sim,rnorm(n=sim.size,mean=xpoints[j],sd=sim.asd))
	}
	sim.sd <- sd(sim)
	sim.hc <- fhc(sim,mc)
	diff.sd <- abs(sim.sd-data.sd)
	diff.bw <- abs(sim.hc-data.hc)
	if((diff.sd<=error.sd)&&(diff.bw<=error.bw)) {
		keep.going <-FALSE
	}
	#cat('OUT',(diff.sd>error.sd),(diff.bw>error.bw),'\n')
}
sim.df <- density(sim,sim.hc)
sim.mean <- mean(sim)
```

```
	if(plots) {
		#par(mfrow=c(1,2))
		plot(data.df,pch='.',main='Data')
		abline(v=data.mean,col='blue')
		abline(v=xpoints,col='red')
		lines(sim.df,pch='.',main='Sim',col='red')
		#abline(v=data.mean,col='blue')
		#abline(v=xpoints,col='red')
		#par(mfrow=c(1,1))
	}

	return(sim)
}
```