A Brief Introduction to Bandwidth Selection Methods for Univariate Kernel Density Estimation

Brad Stieber

Abstract

Univariate kernel density estimation allows an analyst to get a quick glimpse of some of the distributional properties for some data vector they have in hand. Kernel density estimation can also allow a statistician to generate simulations using the estimated density. Of utmost importance in kernel density estimation is selecting the bandwidth, which controls the smoothness of the estimate. Insufficient smoothing leads to an estimate which may detect spurious modality, and over-smoothing may obscure important patterns within the data. In this report, we discuss various methods for selecting an optimal bandwidth for univariate kernel density estimation. We discuss Silverman's rule of thumb (1986), a cross-validation technique, Sheather and Jones' plug-in estimator (1991), and Terrell's maximal smoothing principle (1990). We also visualize results from univariate density estimation using the different bandwidths for various data vectors, referencing a function written in the R computing environment.

Introduction

Kernel density estimation allows a data analyst to get a view of an estimate of the density from which their data may have come. The univariate kernel density estimate can be used in a visualization setting or in a simulation setting. Furthermore, the KDE can present an analyst with a more refined view of the potential modality within their data than a histogram can.

We can use a kernel density estimate for exploratory analysis, or for more thorough simulation studies. The focus of this report is on the bandwidth of a kernel density estimate. The bandwidth controls how smooth our estimate of the underlying density is. If we select a large bandwidth, we will smooth out some of the interesting features of the distribution of the data. If we select a small bandwidth, our density estimate may be too "wiggly", and over exaggerate random perturbations in the data as multi-modality. While many methods exist for selecting a bandwidth, it should be noted that no universally best method has been found (Givens and Hoeting 2013, 332).

Suppose we have a sample $x_1, x_2, \ldots, x_n$ which are i.i.d. observations from some density f. We wish to estimate f using $\hat{f}$ for an analysis. We can use the kernel density estimator $\hat{f}$ to approximate the density.

The kernel density estimate of f is

$$\widehat{f_h}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where h is the smoothing parameter known as the bandwidth, and K is a kernel function which satisfies $\int K(x)dx = 1$. In general, K should also satisfy the conditions (Sheather 2004)

$$\int yK(y)dy = 0$$

$$\int y^2K(y)dy = \mu_2(K) > 0.$$

The kernel function is generally chosen to be a unimodal probability density function that is symmetric around zero. Popular choices include the Gaussian kernel ($K(x) = \phi(x)$), the Epanechnikov kernel ($K(x) = \frac{3}{4}(1 - x^2) * 1(|x| < 1)$), and the triangular kernel ($K(x) = (1 - |x|) * 1(|x| < 1)$). Provided that the kernel

function $K(x)$ satisfies the aforementioned conditions, the choice of kernel is not as important as selecting the bandwidth.

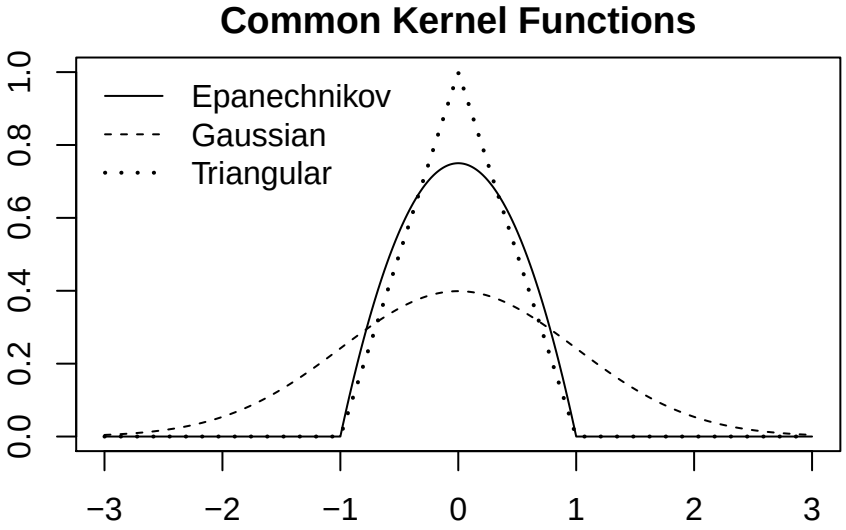


Figure 1: Three Common Kernel Functions: the solid line is the Epanechnikov kernel, the dashed line is the standard Gaussian kernel, and the dotted line is the Triangular kernel

The selection of the bandwidth is more important than the selection of a kernel function. We use the bandwidth to control how much weight we apply to observations around the x for which we are evaluating $\widehat{f_h}(x)$. If h is too small, we assign density too locally, resulting in wiggly estimates; however, if we select h to be too large, we spread density too diffusely, smoothing out interesting features of the data (Givens and Hoeting 2013). We have illustrated the effect of bandwidth in Figure 2, where we have evaluated $\widehat{f_h}(x)$ for 200 points from the $N(2, 3)$ density.

We have provided three density estimates in Figure 2, each using a Gaussian kernel, but different bandwidths. The lighter solid line is too small of a bandwidth ($h = 0.2$), it is too sensitive to local perturbations. The dashed line is too large of a bandwidth ($h = 2$), so it over-smooths the interesting components of the distribution. The heavy line ($h = 0.6$) seems to be a fairly respectable bandwidth, it does not over-smooth the data, but it also highlights some of the interesting features of the data.

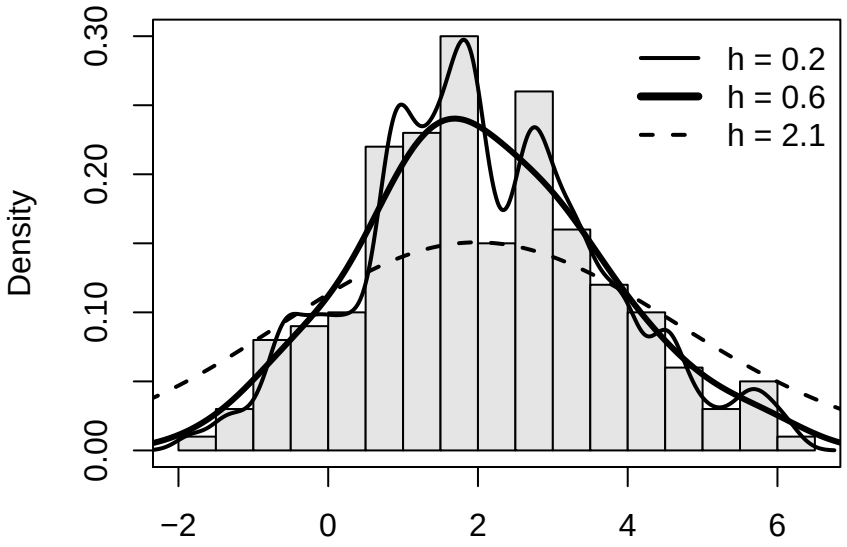


Figure 2: Kernel density estimates using 3 bandwidths for 500 points from N(2, 3)

Methods for evaluating KDE

To calculate $\widehat{f_h}(x)$, we need to compute an estimate for every observation x_i, based on the $n-1$ remaining observations in the data. This computation requires $n*(n-1)$ computations to calculate the final estimate. To get around such a computationally intensive calculation, possible techniques such as linear binning and Fast Fourier Transforms are used (both are implemented in `R`'s `density` function).

Computing the kernel density estimator amounts to piling up mass around each x, then summing the mass up over $x_1, x_2, \ldots, x_n$. Figure 3 demonstrates this computation. We start with 10 i.i.d. observations from $N(0,1)$, which are the red triangles in the plot. Using a Gaussian kernel, we compute the densities (scaled by $1/nh$) for each x, which are the blue curves in the plot. We then sum across these computations and are left with our final KDE, which is the black curve. We have provided plots for a large bandwidth and a small bandwidth, demonstrating the micro and macro effects of bandwidth selection.

Evaluating the Fit of $\widehat{f}_h(x)$

Along with the computation of $\widehat{f_h}(x)$, it is also important to assess the quality of $\hat{f}$ as an estimator of f. One choice would the be the Integrated Squared Error (ISE):

$$ISE(h) = \int_{-\infty}^{\infty} \left(\hat{f}(x) - f(x)\right)^2 dx.$$

ISE calculates the squared error for *this* sample. It might be reasonable to average ISE over *all* samples, resulting in Mean Integrated Squared Error (MISE): $MISE(h) = E\left(ISE(h)\right)$.

We can rewrite $MISE(h)$ as:

$$MISE(h) = \int_{-\infty}^{\infty} MSE_h\left(\hat{f}(x)\right) dx,$$

and decompose MSE as $MSE = bias^2 + var$:

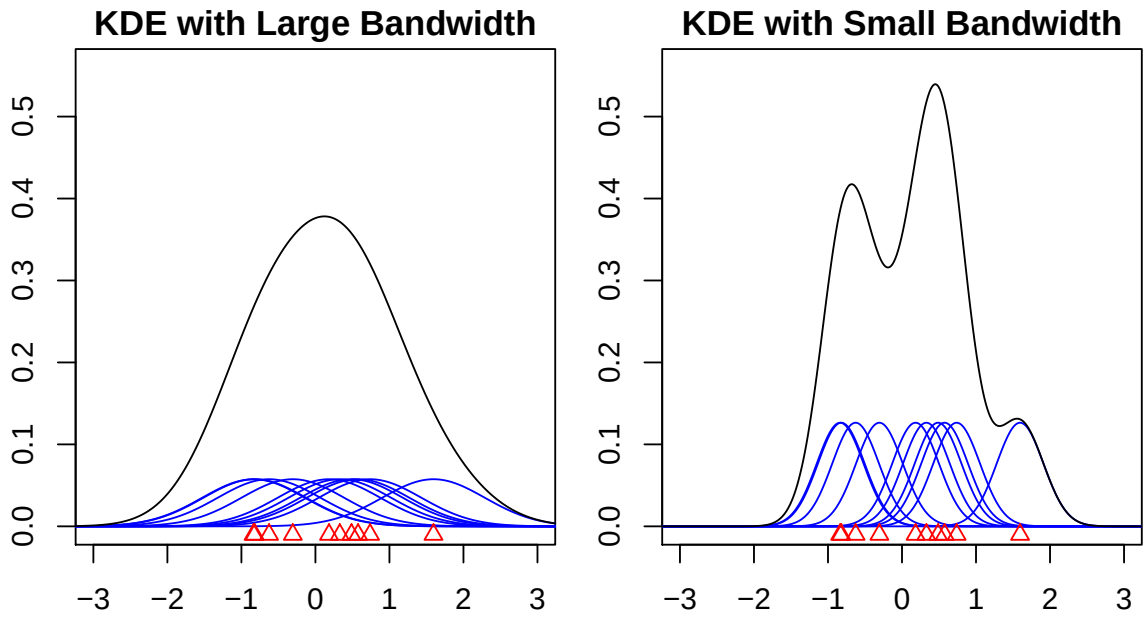


Figure 3: Demonstration of KDE computation. Red triangles are 10 i.i.d. data points from N(0,1). Blue lines are scaled densities at each data point. Black line is final KDE. Left panel uses a large bandwidth (more smoothing), right panel uses a small bandwidth (less smoothing).

$$MISE(h) = \int_{-\infty}^{\infty} var\left(\widehat{f_h}(x)\right) + bias\left(\widehat{f_h}(x)\right)^2 dx.$$

Givens and Hoeting (2013, 327) say of the distinction between ISE and $MISE$[1]:

> The distinction is essentially one between the statistical concepts of loss and risk. Using ISE(h) is conceptually appealing because it assesses the estimator's performance with the observed data. However, focusing on MISE(h) is an effective way to approximate ISE-based evaluation while reflecting the sensible goal of seeking optimal perfomance on average over many data sets.

Givens and Hoeting (2013, 331) derived $E[\widehat{f_h}(x)]$ as:

$$E[\widehat{f_h}(x)] = f(x) + \frac{1}{2}\sigma_K^2 f^{''}(x) + o(h^2),$$

where σ_K^2 is the variance of K ($\int z^2 K(z)dz$). So that $\left(bias(\widehat{f_h}(x))\right)^2 = \frac{1}{4}\sigma_K^4 (f^{''})^2 + o(h^4)$, and integrating $bias^2$ results in:

$$\int bias\left(\hat{f}(x)\right)^2 dx = \frac{1}{4}h^4\sigma_K^4 R(f^{''}) + o(h^4),$$

where R is a measure of the roughness of a function g:

$$R(g) = \int_{-\infty}^{\infty} g^2(z)dz.$$

[1]The reader may be interested in our focus on squared error, instead of opting for the L_1 norm. While the L_1 norm is theoretically appealing, due to its invariance under one-to-one smooth transformations, it can be somewhat difficult to analyze mathematically. For this reason it is neglected here.

It can also be shown that $var\left(\hat{f}(x)\right) = \frac{1}{nh}f(x)R(K) + o(\frac{1}{nh})$, and integrating var results in:

$$\int var\left(\hat{f}(x)\right)dx = \frac{R(K)}{nh} + o(\frac{1}{nh}).$$

Adding $bias^2$ and var together results in:

$$\begin{aligned} MISE(h) &= \frac{R(K)}{nh} + o\left(\frac{1}{nh}\right) + \frac{1}{4}h^4\sigma_K^4 R(f^{''}) + o\left(h^4\right) \\ &= AMISE(h) + o\left(\frac{1}{nh} + h^4\right), \end{aligned}$$

where $AMISE$ is the Asymptotic Mean Integrated Squared Error. If $nh \to \infty$ and $h \to 0$ as $n \to \infty$, then $MISE(h) \to 0$.

Inspection of the $bias^2$ and var components of $MISE(h)$ with respect to the bandwidth h provides an example of the bias-variance tradeoff. Selecting a small value of h increases variance by producing a wiggly estimate, but reduces bias, while selecting a large value of h reduces the variance of the estimator, but increases the bias due to over-smoothing. To find an optimal value, we must balance the bias and variance components of $MISE$, a typical theme in statistical inference.

Finding an optimal bandwidth h_{AMISE}, is straightforward using calculus (ignoring $o\left(\frac{1}{nh} + h^4\right)$):

$$\begin{aligned} AMISE &= \frac{R(K)}{nh} + \frac{h^4\sigma_K^4 R(f^{''})}{4} \\ \frac{dAMISE}{dh} &= \frac{-R(K)}{nh^2} + h^3\sigma_K^4 R(f^{''}) \\ 0 &= \frac{-R(K)}{n} + h^5\sigma_K^4 R(f^{''}) \\ h^5 &= \frac{R(K)}{n\sigma_K^4 R(f^{''})} \\ h_{AMISE} &= \left(\frac{R(K)}{n\sigma_K^4 R(f^{''})}\right)^{\frac{1}{5}}. \end{aligned}$$

All of the components of h_{AMISE} are known to us, except for $R(f^{''})$. Of $R(f^{''})$, Sheather (2004, 589) states:

> Thus, the functional $R(f^{''})$ is a measure of the underlying roughness or curvature. In particular, the larger the value of $R(f^{''})$ is, the larger is the value of AMISE (i.e., the more difficult it is to estimate f) and the smaller is the value of h_{AMISE} (i.e., the smaller the bandwidth needed to capture the curvature in f).

Bandwidth Selection

The possible choices for bandwidth depend on navigating around the fact that $R(f^{''})$ is unknown. We investigate four bandwidth selection strategies which employ different methods to circumvent not knowing $R(f^{''})$.

- Cross validation: choose a different objective function based on ISE, and use a LOO cross validation estimator to find $\int \hat{f}_h f$
- Silverman's ROT: replace f with the $N(0, \sigma^2)$ density
- Sheather-Jones: empirically estimate $f''(x)$ to find h in a two-step method

- Terrell's Maximal Smoothing: minimize $R(f'')$ for a given σ

Unbiased Cross Validation

We can represent integrated squared error as:

$$ISE(h) = \int \left(\hat{f}_h - f\right) = \int \hat{f}_h^2 - 2\int \hat{f}_h f + \int f^2.$$

In $ISE(h)$, only the first two terms rely on h, and the first term is entirely known. We can estimate the second term ($2\int \hat{f}_h f = 2E(\hat{f}_h)$) using $2/n \sum_{i=1}^{n} \hat{f}_{-i}(x_i)$. Where

$$\hat{f}_{-i}(x_i) = \frac{1}{h(n-1)} \sum_{j \neq i} K\left(\frac{x_i - x_j}{h}\right)$$

is the leave-one-out kernel density estimator using all of the data except x_i.

We then attempt to minimize

$$UCV(h) = R(\hat{f}) - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{-i}(x_i),$$

with respect to h.

$UCV(h)$ is called the "unbiased cross-validation" criterion because $E\left[UCV(h) + R(f)\right] = MISE(h)$ (Givens and Hoeting 2013, 333). To estimate the unbiased cross validation bandwidth for a data vector `x` in `R`, a user can write `bw.ucv(x)`. Along with "unbiased cross-validation", it is sometimes referred to as "least-squares cross validation" as it minimizes the integrated squared error. When we refer to the cross-validated bandwidth selector, we denote the bandwidth as h_{LSCV}.

For this report, we have focused on the Gaussian kernel. $UCV(h)$ has a straightforward expression (equation 10.23, Givens and Hoeting (2013, 333)) when a Gaussian kernel is used, which makes the computation of h_{LSCV} rather tidy.

The unbiasedness of the objective function is a nice feature, but it comes at the cost of imbuing h_{LSCV} with excessive variance (Jones, Marron, and Sheather 1992, 6). The objective function UCV tends to depend too heavily on the sample in hand, since it is still random (unlike $MISE$, which is averaged over *all* samples). The heavy dependence on the data in hand often leads to high variation in $\hat{f}_h$. This excessive variation is displayed in the *Examples* section of this report.

Silverman's Rule of Thumb

A simpler method relies on selecting an appropriate density to approximate $R(f'')$ in

$$h_{AMISE} = \left(\frac{R(K)}{n\sigma_K^4 R(f'')}\right)^{\frac{1}{5}}.$$

If we choose f to be the normal density with mean 0 and variance σ^2, and use set K to be the Gaussian kernel, we can express h_{AMISE} as:

$$h_{AMISE} = \left(\frac{(2\sqrt{\pi})^{-1}}{\frac{3}{8}\pi^{-\frac{1}{2}}\sigma^{-5}} \right)^{1/5} n^{-1/5}$$
$$= 1.06\sigma n^{-\frac{1}{5}}.$$

A natural estimate of σ is $\hat{\sigma}$, the sample standard deviation. If the true f deviates from the features of a normal distribution (for instance, it is multimodal), this approximation may do a poor job (i.e. it will oversmooth). A possiblity is to use the interquartile range, which is a more robust measure of spread. We then replace $\hat{\sigma}$ with $\tilde{\sigma} = min\left[\hat{\sigma}, \frac{IQR(x)}{1.34}\right]$. This leads to Silverman's rule of thumb:

$$h_{SROT_1} = 1.06\tilde{\sigma}n^{-\frac{1}{5}}.$$

In our report, we replace 1.06 in the previous equation with 0.9, so that $h_{SROT} = \frac{0.9}{1.06}h_{SROT_1}$. This replacement dampens some of the oversmoothing that takes place when this method is used, and attempts to avoid missing bimodality in the distribution (Silverman 1986, 48). In `R`, a user can obtain h_{SROT_1} for a data vector `x` by using `bw.nrd(x)`, and can obtain h_{SROT} by using `bw.nrd0(x)`.

The computation of h_{SROT} is fairly simple, and requires no optimzation or cross-validation scheme. The simplicity comes at a cost; however, since this bandwidth tends to oversmooth the estimate, even after the IQR correction is made.

Sheather-Jones Bandwidth

Silverman's rule of thumb relies on selecting a candidate density to estimate $R(f'')$. Sheather and Jones (1991) opt for a different approach: empirically estimate $R(f'')$. An empirical estimate of f'' is formulated as:

$$\hat{f}''(x) = \frac{d^2}{dx^2}\left\{ \frac{1}{nh_0}\sum_{i=1}^{n} L\left(\frac{x - x_i}{h_0}\right)\right\}$$
$$= \frac{1}{nh_0^3}\sum_{i=1}^{n} L''\left(\frac{x - x_i}{h_0}\right).$$

Where h_0 is another bandwidth, different from h_{opt} (and typically $h_0 > h_{opt}$ Givens and Hoeting (2013, 336)), and L is a kernel function.

To compute the optimal bandwidth h_{SJ}, Sheather and Jones use a two-step process. First, a simple rule of thumb is used to calculate h_0. Then h_0 is used to estimate $R(f'')$, which is the only unknown quantity in the expression for h_{AMISE}. After plugging in the estimator for $R(f'')$ in the expression for h_{AMISE}, we arrive at the Sheather-Jones bandwidth. An expression for the optimal bandwidth when $L = \phi$ is in the Appendix. This expression requires an root-finding scheme such as Newton's method.

In simulation studies (summarized in Jones, Marron, and Sheather (1996)), it was seen that h_{SJ} tended to be superior to both h_{LSCV} and h_{SROT}. In general, h_{SJ} demonstrates less variation than h_{LSCV} and is centered near h_{AMISE} for easy-to-estimate densities. For more difficult densities, h_{SJ} will overshoot h_{AMISE}, but is still much smaller than h_{SROT}. h_{SJ}, then, seems to be a suitable compromise between the low-variance high-bias h_{SROT} and the high-variance low-bias h_{LSCV}.

Terrell's Maximal Smoothing Principle

Terrell (1990) opted for a different approach in determining an optimal bandwidth. Rather than focusing on finding a good estimator of f'' or $R(f'')$ in the expression for h_{AMISE}, Terrell chose to find some density function g that would minimize $R(g'') = \int (g''(x))^2 dx$ for a given variance σ^2. By minimizing $\int (g'')^2$, we

could determine an upper bound for h_{opt}, by using g as our estimate for f. Terrell (1990, 472) justifies using the maximally smoothed bandwidth, because it avoids the tendency of the undersmoothed density estimate to "display features such as asymmetries and multiple modes that could have come about by chance".

Terrell's estimate is built on the result that the $beta(k+2, k+2)$ family minimizes $\int (f^{(k)})^2$ for a given standard deviation (Terrell 1990, 471). For a kernel with variance 1 (e.g. the Gaussian kernel), the maximal smoothing bandwidth is:

$$h_{MS} = 3\hat{\sigma}\left(\frac{R(K)}{35n}\right)^{\frac{1}{5}}.$$

The maximal smoothing principle has an intuitive justification, and its ideas permeate through wider areas of statistical inference. It would be desirable for an inferential scheme to prevent an analyst from claiming features of a data set exist that may have arisen by pure chance alone (Terrell 1990, 472). The maximal smoothing principle applies this idea to the display of kernel density estimates.

In terms of kernel density estimation; however, the maximal smoothing principle provides only an upper bound on the optimal bandwidth. Silverman's rule of thumb is an over-smoother, and Terrell's maximal smoothing bandwidth smooths out even more features of the data (Jones, Marron, and Sheather 1992, 9) than h_{SROT} does.

Examples Using the `kde` Function

To examine the four different bandwidth selections we have investigated, we use four different data vectors to investigate the changes in density estimation as the data deviates from the normal distribution. Two vectors are simulated, and two vectors are from real data sets.[2] Our main interest is on if the bandwidth options lead to undersmoothing or oversmoothing of the data.

We plot histograms and density estimates for each of the data vectors using the `kde` function (source code available in *Appendix*). The `kde` function takes in a data vector, and calculates h_{SROT}, h_{SJ}, h_{LSCV}, and h_{MS}. It then calculates $\widehat{f_h}(x)$. For small data vectors ($n \leq 1,000$), `kde` will calculate the densities "by hand", using a grid of size 512. For larger data vectors, the function will use the `density` function in base `R`. In addition to creating plots, the `kde` function will print the four bandwidths, and can also return $\widehat{f_h}(x)$ for an evenly spaced grid of length 512 for each h.

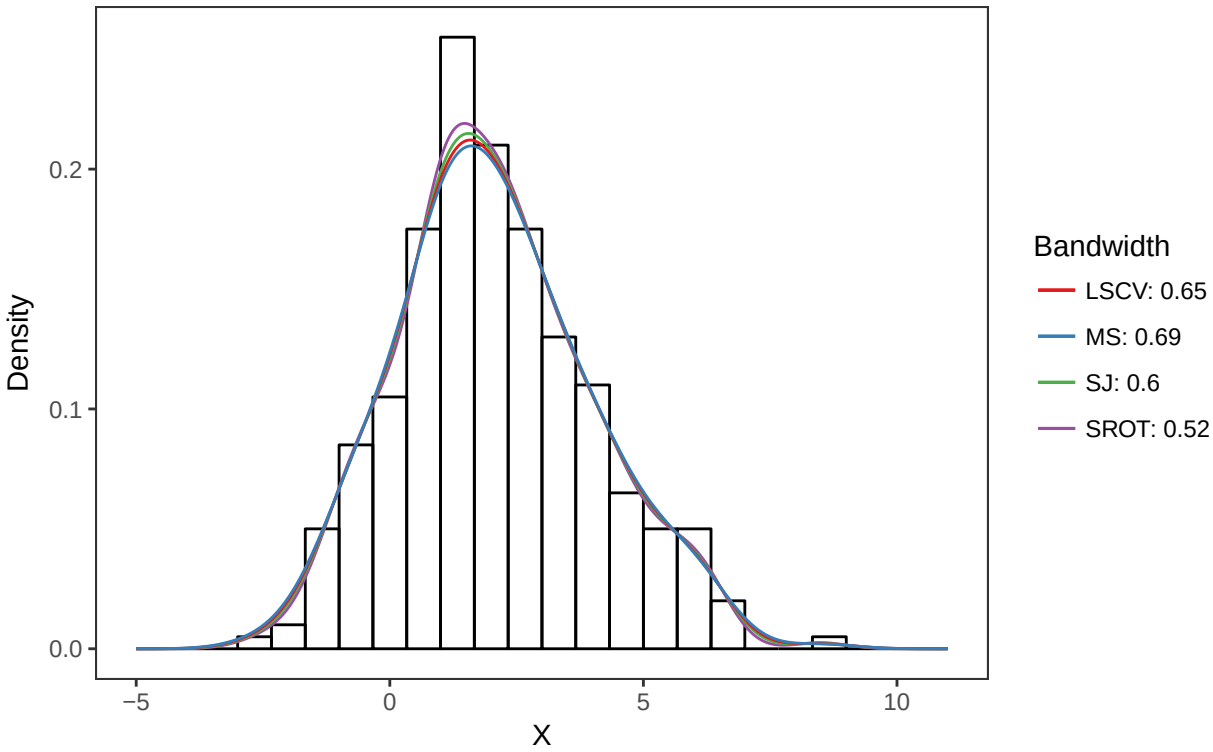
In each example, we choose a data vector and display the output from `kde`. We then attempt to interpret the results, taking special care to analyze the effect of each bandwidth choice on $\hat{f}_h$. Inspecting a kernel density estimate is not an objective task, one analyst will perceive different features than another. It is important to present multiple density estimates, and iterate through different bandwidths to arrive at a more common conclusion.

We include each $\hat{f}_h$ on the same graph rather than allowing each $\hat{f}_h$ to have its own panel. This strategy allows for the other three $\hat{f}_h$'s to act as "reality checks" for the $\hat{f}_h$ of the h we are focusing on. Keeping the estimates on the same graph allows for effective comparisons between the $\hat{f}_h$'s.

[2] The `R` code used to generate the data can be found on the author's Github: https://github.com/bgstieber/Project771/blob/master/Project%20Files/DataGeneration.R

Example 1: 300 samples from $N(2, 2^2)$

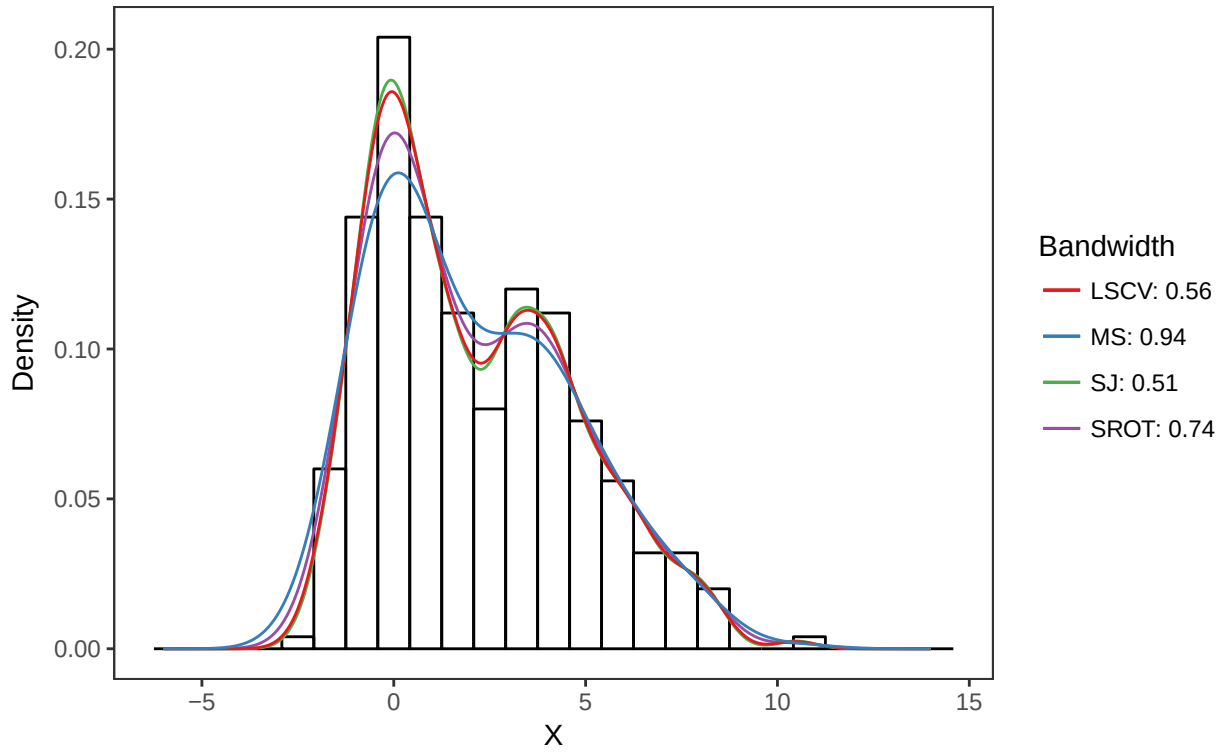
```
##          LS CV           T MS          SJ BW Silverman's ROT
##          0.648          0.691          0.598          0.519
```



We begin by examining $\widehat{f}_h(x)$ for a data vector sampled solely from $N(2, 2^2)$. In this case, we see that each bandwidth selection does a fairly good job of estimating the density. h_{SROT} relies on a normal approximation, so we would expect this bandwidth to perform well on this vector. In fact, it is the most aggressive bandwidth estimator, resulting in the smallest h value of the four. A *best bandwidth* is not apparent from these results.

Example 2: 50/50 Mixture of $N(0,1)$ and $N(4,2^2)$

```
##          LS CV           T MS        SJ BW Silverman's ROT
##          0.562          0.944          0.515          0.742
```

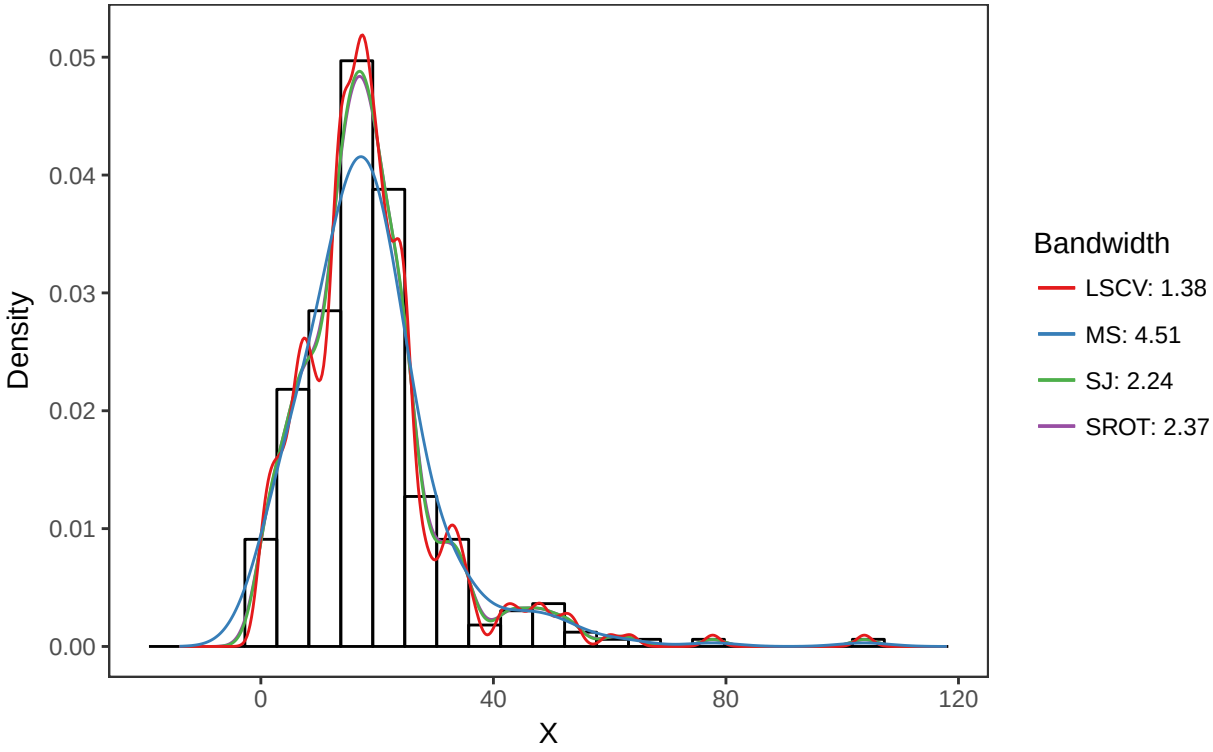


Moving on to a somewhat harder density to estimate, we simulate a 50/50 distribution coming from $N(0,1)$ and $N(2,2^2)$. h_{SJ} and h_{LSCV} both seem to capture the bimodality of the distribution quite well. h_{SROT} smooths out some of the interesting features, but the bimodality is still somewhat detectable. h_{MS} oversmooths disastrously - without knowing the underlying structure of the data, it would not be obvious that the data came from a bimodal distribution.

Example 3: Mixture of $\Gamma(\alpha = 1, \beta = 1/20)$, $\Gamma(\alpha = 4, \beta = 1/5)$, $\Gamma(\alpha = 20, \beta = 1)$

A mixture of gamma distributions may seem like a "toy" example, but the gamma distribution can be very helpful in modeling rainfall (e.g. Husak, Michaelsen, and Funk (2007)).

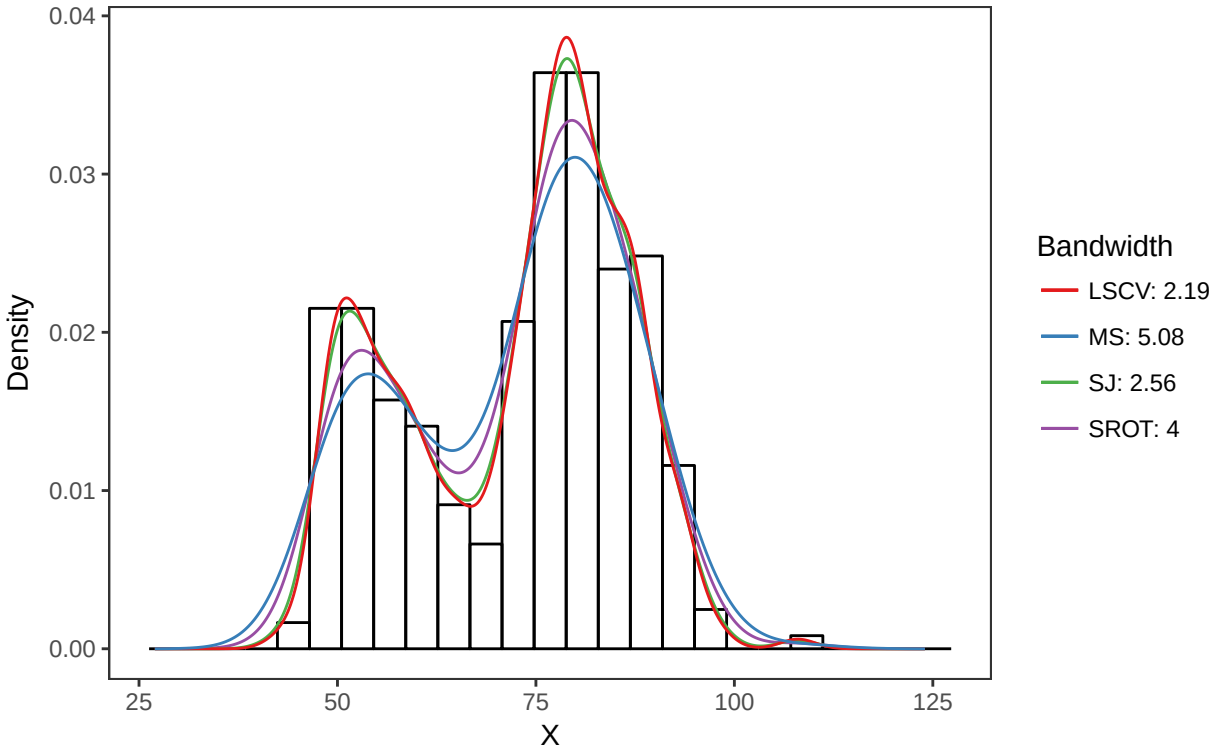
```
##          LS CV           T MS       SJ BW Silverman's ROT
##          1.382          4.514          2.241          2.374
```



Here we see the excessive variation caused by using h_{LSCV}. The density estimate provided by h_{LSCV} is far too wiggly to be useful for a data analyst. We also see that h_{MS} provides a much smoother estimate than the those proposed by Silverman and Sheather and Jones, it seems that it is too conservative. The best bandwidth would take a value somewhere in the range between h_{SJ} and h_{MS}.

Example 4: Old Faithful Waiting Time

```
##          LS CV           T MS        SJ BW Silverman's ROT
##          2.190          5.081        2.560          3.998
```



Using data from Azzalini and Bowman (1990), contained in the **MASS** package, we calculate the kernel density estimate for the waiting time for eruptions from August 1 to August 15, 1985 at the Old Faithful geyser in Yellowstone National Park, Wyoming.

Each bandwidth choice seems to capture the bimodality in the distribution of waiting times with varying levels of success. h_{LSCV} may be too aggressive, while h_{MS} does not reach far enough up the left-side peak. h_{SJ} and h_{LSCV} seem grouped together and h_{SROT} and h_{MS} seem grouped together as well. Between h_{SJ} and h_{LSCV}, we would want to choose the more conservative bandwidth, which is h_{SJ}. Conversely, between h_{MS} and h_{SROT}, we would want to choose the more aggressive bandwidth, which is h_{SROT}.

Conclusion

In this report we have presented four options for selecting a bandwidth for univariate kernel density estimation. We have discussed bandwidths that oversmooth (h_{MS}), and bandwidths that undersmooth (h_{LSCV}). No universally optimal bandwidth selection has been found, which makes selecting a bandwidth an iterative and subjective process.

A safe option for a novice data analyst is to use at least two bandwidths when analyzing the data. One bandwidth should have the tendency to undersmooth (e.g. h_{SROT} or h_{MS}), and another bandwidth should produce estiamtes that are more wiggly (e.g. h_{SJ}, **but not** h_{LSCV}). If vizualizing is the goal (rather than simulation), then including a histogram will provide even more clarity when making density plots (note that making a histogram also requires selecting a bandwidth, though).

In our *Examples* section, we saw the devastating effects of oversmoothing and undersmoothing. There is a large amount of literature on univariate bandwidth selection, but a universally optimal bandwidth has

yet to be found. Selecting a bandwidth results presents a bias-variance tradeoff to the analyst, with a good bandwidth being depedent on the analytical task. Sheather (2004, 596) suggests to "produce a family of density estimates based on a number of values of the bandwidth". We agree with this suggestion, and believe it to be sage advice for exploratory analysis.

Question

Using the `density` function and the beer data (`abv.complete`, which are the alcohol by volume measures for the top 250 beers as rated by BeerAdvocate.com) provided below, estimate a 95% pointwise bootstrapped percentile interval. Additionally, use the approximate variance function $\tilde{V}$ given by Jann (2007, 2) to estimate a 95% pointwise confidence interval. Create plots using `bw.SJ` and `bw.nrd0`.

$$\tilde{V}\left(\widehat{f_K}(x;h)\right) = \frac{1}{nh}R(K)\widehat{f_K}(x;h) - \frac{1}{n}\widehat{f_K}(x;h)^2$$

```
# data comes from the ABV of beer advocate's top 250 beers for 2016
# 2 beers have missing abv data
# quick R script to read in the .csv
library(RCurl)
git_file <-
getURL("https://raw.githubusercontent.com/bgstieber/Top250Beer/master/Top250Beers.csv")
beer_data <- read.csv(text = git_file, stringsAsFactors = FALSE)
abv.complete <- beer_data[!is.na(beer_data$ABV), ]$ABV
```

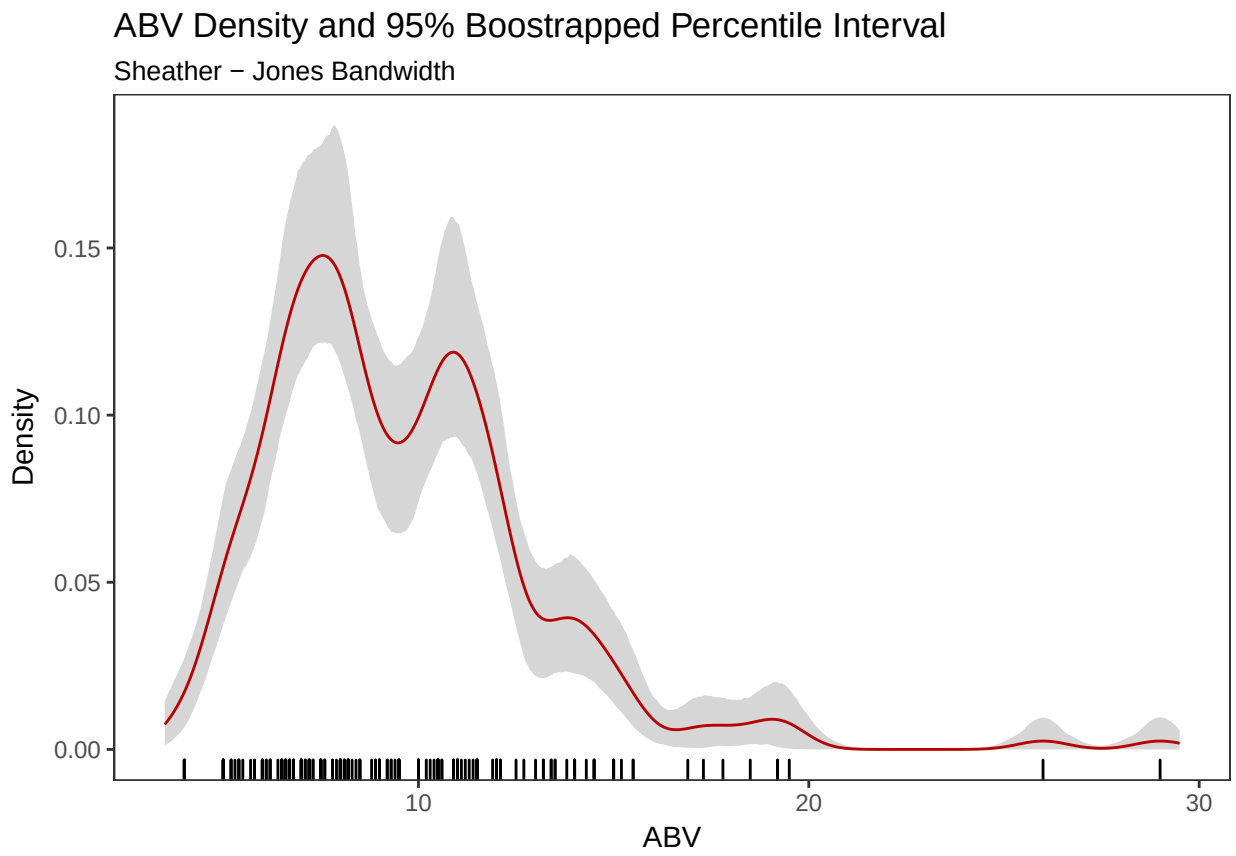
Solution

Bootstrap percentile interval

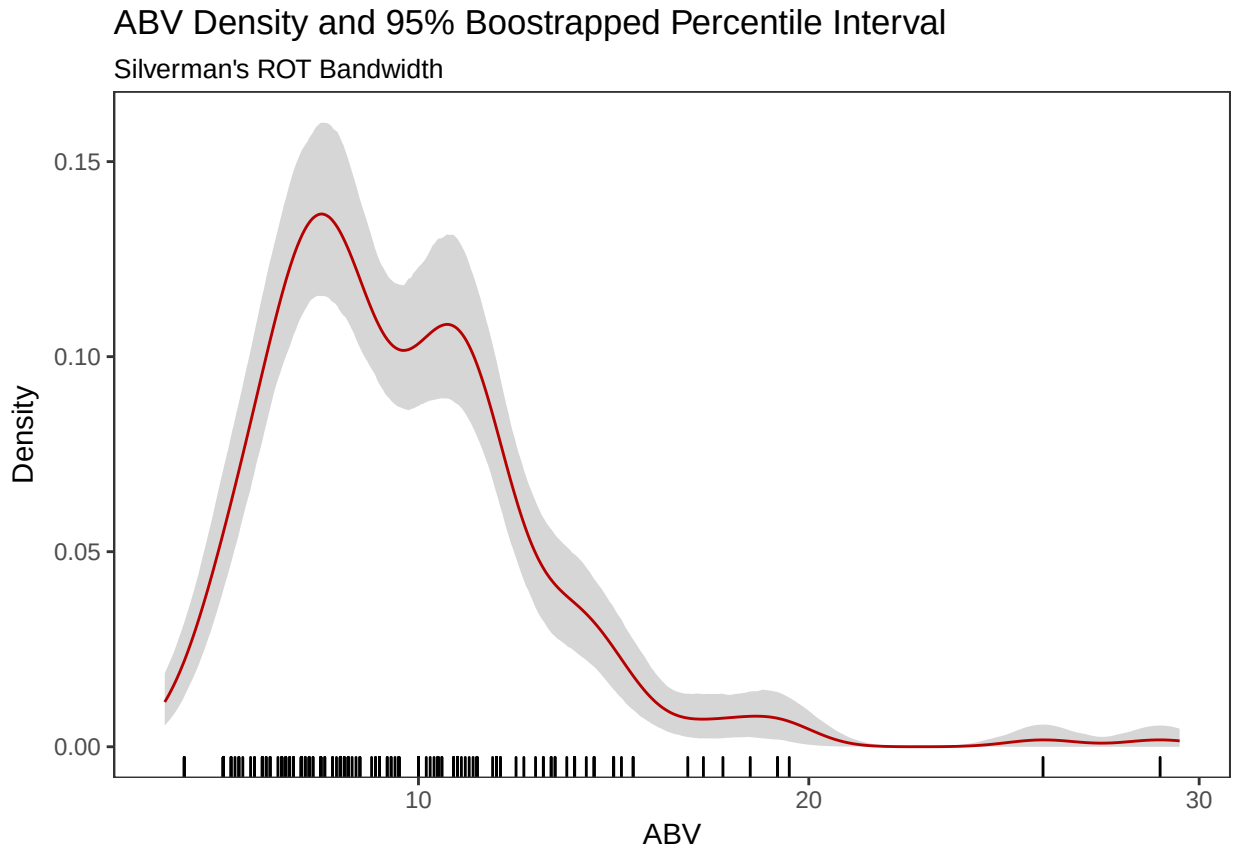
```
#function that computes the 95% pointwise percentile interval
boot_dens <- function(x, from, to, B, bw){


    orig_dens <- density(x, from = from, to = to, bw = bw)

    boot_dens <- matrix(0, ncol = B, nrow = length(orig_dens$x))

    for(i in 1:B){
        boot.x <- sample(x, length(x), replace = T)
        boot.density <- density(boot.x, from = from, to = to, bw = bw)
        boot_dens[,i] <- boot.density$y
    }

    lower <- apply(boot_dens, 1, function(mat) quantile(mat, .025))
    upper <- apply(boot_dens, 1, function(mat) quantile(mat, .975))

    data.frame(grid = orig_dens$x,
               yhat = orig_dens$y,
               lower = lower,
               upper = upper)
```

```
}
```

```
range_abv <- range(abv.complete)
range_abv <- range_abv + c(-.5, .5) #wiggle room

set.seed(1984)

boot.abv.SJ <- boot_dens(x = abv.complete, from = range_abv[1],
                         to = range_abv[2], B = 1500, bw = 'SJ')

boot.abv.nrd0 <- boot_dens(x = abv.complete, from = range_abv[1],
                         to = range_abv[2], B = 1500, bw = 'nrd0')

ggplot(boot.abv.SJ, aes(x = grid))+
    geom_ribbon(aes(ymin = lower, ymax = upper),
                alpha = .2)+
    geom_line(aes(y = yhat),
              colour = '#b70101')+
    geom_rug(data = data.frame(abv.complete),
             aes(x = abv.complete, y = 0.001),
             inherit.aes = FALSE, sides = 'b')+
    xlab('ABV')+ylab('Density')+
    ggtitle('ABV Density and 95% Boostrapped Percentile Interval',
            subtitle = 'Sheather - Jones Bandwidth')
```



```
ggplot(boot.abv.nrd0, aes(x = grid))+
    geom_ribbon(aes(ymin = lower, ymax = upper),
                alpha = .2)+
    geom_line(aes(y = yhat),
              colour = '#b70101')+
    geom_rug(data = data.frame(abv.complete),
             aes(x = abv.complete, y = 0.001),
             inherit.aes = FALSE, sides = 'b')+
    xlab('ABV')+ylab('Density')+
    ggtitle('ABV Density and 95% Boostrapped Percentile Interval',
            subtitle = "Silverman's ROT Bandwidth")
```

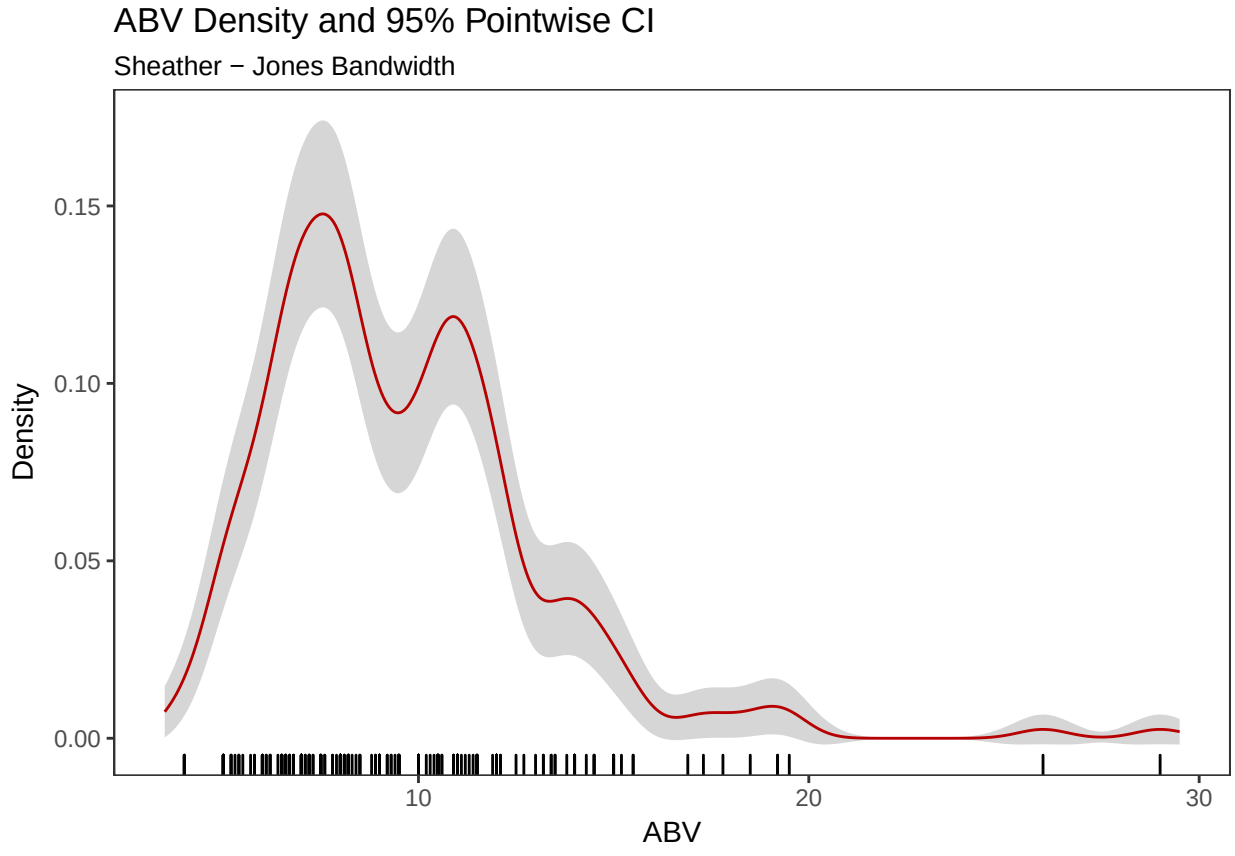


Approximate Confidence Interval

```
app_var <- function(x, bw){
    R <- 1 / (2 * sqrt(pi)) # assume gaussian
    h <- ifelse(bw == 'SJ', bw.SJ(x), bw.nrd0(x))
    n <- length(x)
    range_x <- range(x)
    range_x <- range_x + c(-.5, .5) #wiggle room
    dens_x <- density(x, from = range_x[1], to = range_x[2],
                      kernel = 'gaussian', bw = h)

    var.dens <- (1 / (n * h)) * R * dens_x$y - ((1 / n) * (dens_x$y) ^ 2)
```

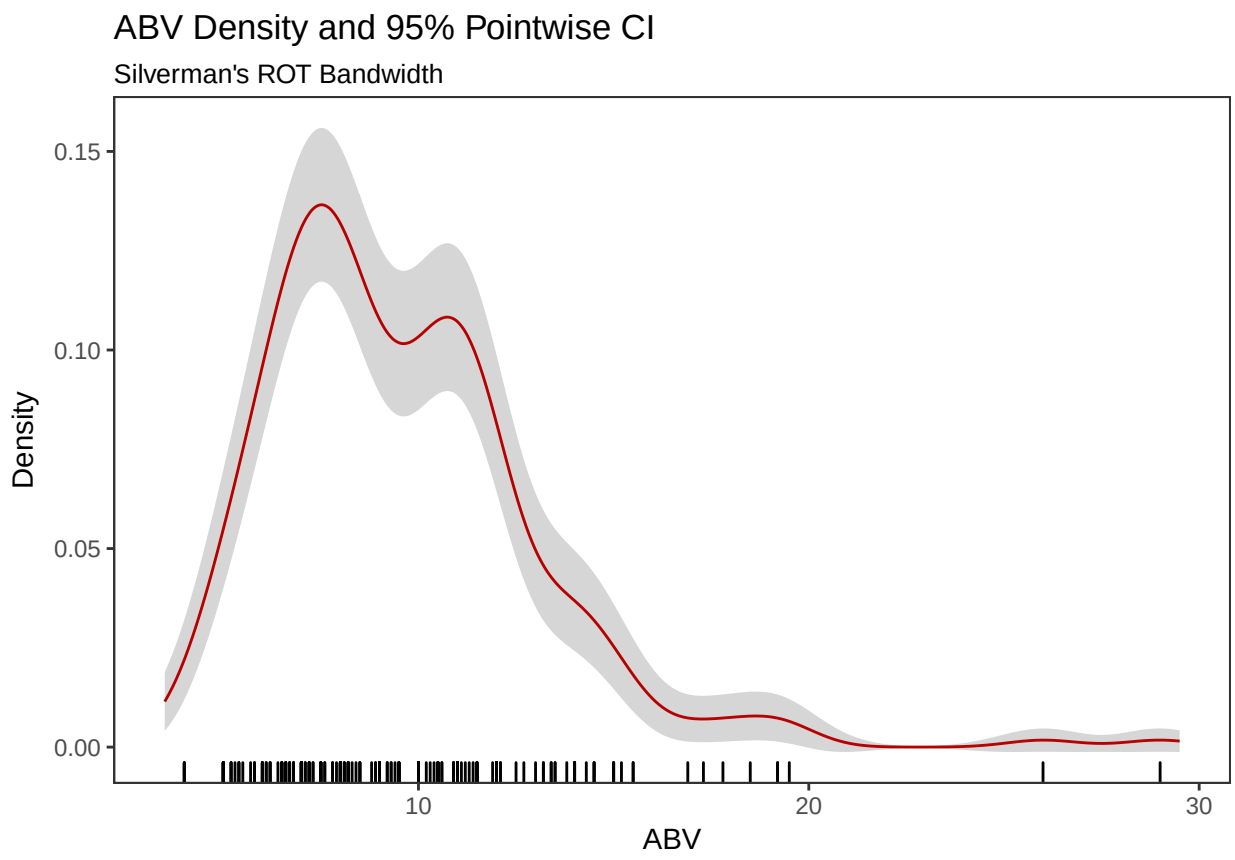
```
    lower <- dens_x$y - (2 * sqrt(var.dens))
    upper <- dens_x$y + (2 * sqrt(var.dens))

    data.frame(grid = dens_x$x,
               yhat = dens_x$y,
               lower = lower,
               upper = upper)
}
```

```
app.abv.SJ <- app_var(x = abv.complete, bw = 'SJ')
app.abv.nrd0 <- app_var(x = abv.complete, bw = 'nrd0')

ggplot(app.abv.SJ, aes(x = grid))+
    geom_ribbon(aes(ymin = lower, ymax = upper),
                alpha = .2)+
    geom_line(aes(y = yhat), colour = '#b70101')+
        geom_rug(data = data.frame(abv.complete),
             aes(x = abv.complete, y = 0.001),
             inherit.aes = FALSE, sides = 'b')+
    xlab('ABV')+
    ylab('Density')+
    ggtitle('ABV Density and 95% Pointwise CI',
            subtitle = 'Sheather - Jones Bandwidth')
```




```
ggplot(app.abv.nrd0, aes(x = grid))+
    geom_ribbon(aes(ymin = lower, ymax = upper),
```

```
              alpha = .2)+
    geom_line(aes(y = yhat), colour = '#b70101')+
        geom_rug(data = data.frame(abv.complete),
             aes(x = abv.complete, y = 0.001),
             inherit.aes = FALSE, sides = 'b')+
    xlab('ABV')+
    ylab('Density')+
    ggtitle('ABV Density and 95% Pointwise CI',
            subtitle = "Silverman's ROT Bandwidth")
```



Appendix

kde function in R

```
Knorm <- function(u) dnorm(u)
kde <- function (x, grid = 512, give_dens = FALSE)
{
    require(dplyr)
    require(ggplot2)

    n <- length(x)

    h.SROT = 0.9 * min(c(IQR(x) / 1.34, sd(x))) * (n)^(-1/5)
```

```
h.LSCV = bw.ucv(x, lower = .05 * h.SROT, upper = 1.2 * h.SROT)
h.SJ = bw.SJ(x)
h.MS = 3 * ((2 * sqrt(pi))^(-1) / (35 * n))^(1/5) * sd(x)
h_vec <- c(h.SROT, h.LSCV, h.SJ, h.MS)

max_h <- max(h_vec)
low_grid <- floor(min(x) - max_h * 3)
high_grid <- ceiling(max(x) + max_h * 3)



print(c('LS CV' = round(h.LSCV, 3),
        'T MS' = round(h.MS, 3),
        'SJ BW' = round(h.SJ, 3),
        "Silverman's ROT" = round(h.SROT, 3)))


if(n > 1000){
    message(paste0('Note: x too large (n = ', n,
                   ' > 1000). Using `density` instead'))

    all_dens <- do.call('cbind',
                        lapply(h_vec, function(bw)
                            density(x = x, from = low_grid, to = high_grid,
                                    bw= bw)$y))


    dens_SNR <- density(x, bw = h.SNR,
                        from = low_grid,
                        to = high_grid)

    sum_results <- data.frame(
        X_grid = dens_SNR$x,
        kde_hSROT = all_dens[,1],
        kde_hSJ = all_dens[,3],
        kde_hLSCV = all_dens[,2],
        kde_hMS = all_dens[,4]
    )

}else{

    grid_vals <- seq(low_grid, high_grid, length.out = grid)

    grid_df <- expand.grid(x = x, grid = grid_vals)



    newx.hsrot <- (grid_df[,2] - grid_df[,1]) / h.SROT
    newx.hsj <- (grid_df[,2] - grid_df[,1]) / h.SJ
    newx.hLSCV <- (grid_df[,2] - grid_df[,1]) / h.LSCV
    newx.hMS <- (grid_df[,2] - grid_df[,1]) / h.MS

    grid_df[,3] <- (1 / (n * h.SROT)) * Knorm(newx.hsrot)
```

```
        grid_df[,4] <- (1 / (n * h.SJ)) * Knorm(newx.hsj)
        grid_df[,5] <- (1 / (n * h.LSCV)) * Knorm(newx.hLSCV)
        grid_df[,6] <- (1 / (n * h.MS)) * Knorm(newx.hMS)


        names(grid_df) <- c(
            "X_act",
            "X_grid",
            'kde_hSROT',
            'kde_hSJ',
            'kde_hLSCV',
            'kde_hMS'
        )


        sum_results <-
            grid_df %>%
            group_by(X_grid) %>%
            summarise(kde_hSROT = sum(kde_hSROT),
                      kde_hSJ = sum(kde_hSJ),
                      kde_hLSCV = sum(kde_hLSCV),
                      kde_hMS = sum(kde_hMS))
    }


    p1 <- ggplot()+
        geom_histogram(data = NULL, aes(x = x, y = ..density..),
                       bins = 1.4 * ceiling(sqrt(n)),
                       fill = NA, colour = 'black')+
        geom_line(data = sum_results, aes(x = X_grid, y = kde_hSROT,
                                          colour = paste('SROT:', round(h.SROT, 2))))+
        geom_line(data = sum_results, aes(x = X_grid, y = kde_hSJ,
                                          colour = paste('SJ:', round(h.SJ, 2))))+
        geom_line(data = sum_results, aes(x = X_grid, y = kde_hLSCV,
                                          colour = paste('LSCV:', round(h.LSCV, 2)))) +
        geom_line(data = sum_results, aes(x = X_grid, y = kde_hMS,
                                          colour = paste('MS:', round(h.MS, 2))))+
        xlab('X') + ylab('Density')+
        theme_bw()+
        theme(panel.grid = element_blank())+
        scale_colour_brewer(palette = 'Set1',
                            name = 'Bandwidth')
    plot(p1)

    if(give_dens){
        return(sum_results)
    }
}
```

Expression for Sheather-Jones Bandwidth

As found in Givens and Hoeting (2013, 337), Sheather-Jones bandwidth can be found by solving the equation (using $L = \phi$):

$$\left(\frac{R(K)}{n\sigma_K^4 \hat{R}_{\hat{\alpha}(h)}(f'')} \right)^{1/5} - h = 0,$$

where:

$$\begin{aligned}
\hat{R}_{\hat{\alpha}(h)}(f'') &= \frac{1}{n(n-1)\alpha^5} \sum_{i=1}^{n} \sum_{j=1}^{n} \phi^{(4)} \left(\frac{x_i - x_j}{\alpha} \right), \\
\hat{\alpha}(h) &= \left(\frac{6\sqrt{2}h^5 \hat{R}_a(f'')}{\hat{R}_b(f''')} \right)^{1/7}, \\
\hat{R}_a(f'') &= \frac{1}{n(n-1)a^5} \sum_{i=1}^{n} \sum_{j=1}^{n} \phi^{(4)} \left(\frac{x_i - x_j}{a} \right), \\
\hat{R}_b(f'') &= \frac{1}{n(n-1)b^7} \sum_{i=1}^{n} \sum_{j=1}^{n} \phi^{(6)} \left(\frac{x_i - x_j}{b} \right), \\
a &= \frac{0.920IQR}{n^{1/7}} \\
b &= \frac{0.912IQR}{n^{1/9}},
\end{aligned}$$

where IQR is the interquartile range of the data, and $\phi^{(i)}$ is the ith derivative of the standard normal density function. To determine the Sheather-Jones bandwidth in `R`, a user can write `bw.SJ(x)`, which uses the `uniroot` function to perform the root-finding.

References

Azzalini, A, and A Bowman. 1990. "A Look at Some Data on the Old Faithful Geyer." *Journal of the Royal Statistical Society* 39 (3): 357–65.

Givens, Geof, and Jennifer Hoeting. 2013. *Computational Statistics*. 2nd ed. John Wiley & Sons.

Husak, Gregory J., Joel Michaelsen, and Chris Funk. 2007. "Use of the gamma distribution to represent monthly rainfall in Africa for drought monitoring applications." *International Journal of Climatology* 27.7 (June 2007): 935–44. doi:10.1002/joc.

Jann, Ben. 2007. "Univariate kernel density estimation." *Statistical Software Component*, no. 3. http://fmwww.bc.edu/RePEc/bocode/k/kdens.pdf.

Jones, M. C., J. S. Marron, and Simon J. Sheather. 1992. "Progress in data-based bandwidth selection for Kernel density estimation." *Computational Statistics* 11 (3): 337–81.

———. 1996. "A Brief Survey of Bandwidth Selection for Density Estimation." *Journal of the American Statistical Association* 91 (433): 401–7. doi:10.1080/01621459.1996.10476701.

Sheather, Simon J. 2004. "Density Estimation." *Stat. Sci.* 19 (4): 588–97. doi:10.1214/088342304000000297.

Sheather, Simon J., and M. C. Jones. 1991. "A Reliable Data-Based Bandwidth Selection Method for Kernel

Density Estimation." *Journal of the Royal Statistical Society* 53 (3): 683–90.

Silverman, Bernard W. 1986. *Density Estimation for Statistics and Data Analysis*. CRC press.

Terrell, George R. 1990. "The Maximal Smoothing Principle in Density Estimation." *Journal of the American Statistical Association* 85 (410): 470–77. doi:10.2307/2289786.