Stat 361 - Recitation 13

Probability Density Estimation

Orçun Oltulu

21 - 22 / 05 / 2020

The Averaged Shifted Histogram:

The optimal bin width does not determine the location of the center or endpoints of the bin, however. For example, using truehist (MASS), we can easily shift the bins from left to right using the argument x_0, while keeping the bin width constant. Shifting the class boundaries changes the density estimates, so several different density estimates are possible using the same bin width.

The Average Shifted Histogram (ASH) proposed by Scott averages the density estimates. That is, the ASH estimate of density is

$$\widehat{f_{ash}}(x) = \frac{1}{m}\sum_{j=1}^{m} \hat{f}_j(x)$$

where the class boundaries for estimate $\hat{f}_{j+1}(x)$ are shifted by h/m from the boundaries for $\hat{f}_j(x)$. Here we are viewing the estimates as m histograms with class width h. Alternately we can view the ASH estimate as a histogram with widths h/m. The optimal bin width for the naive ASH estimate of a Normal(μ, σ^2) density is

$$h = 2.576 * \sigma * n^{-1/5}$$

- An Average Shifted Histogram (ASH) constructs a density estimate by averaging together estimates produced by a set of shifted histograms.

Question 1:

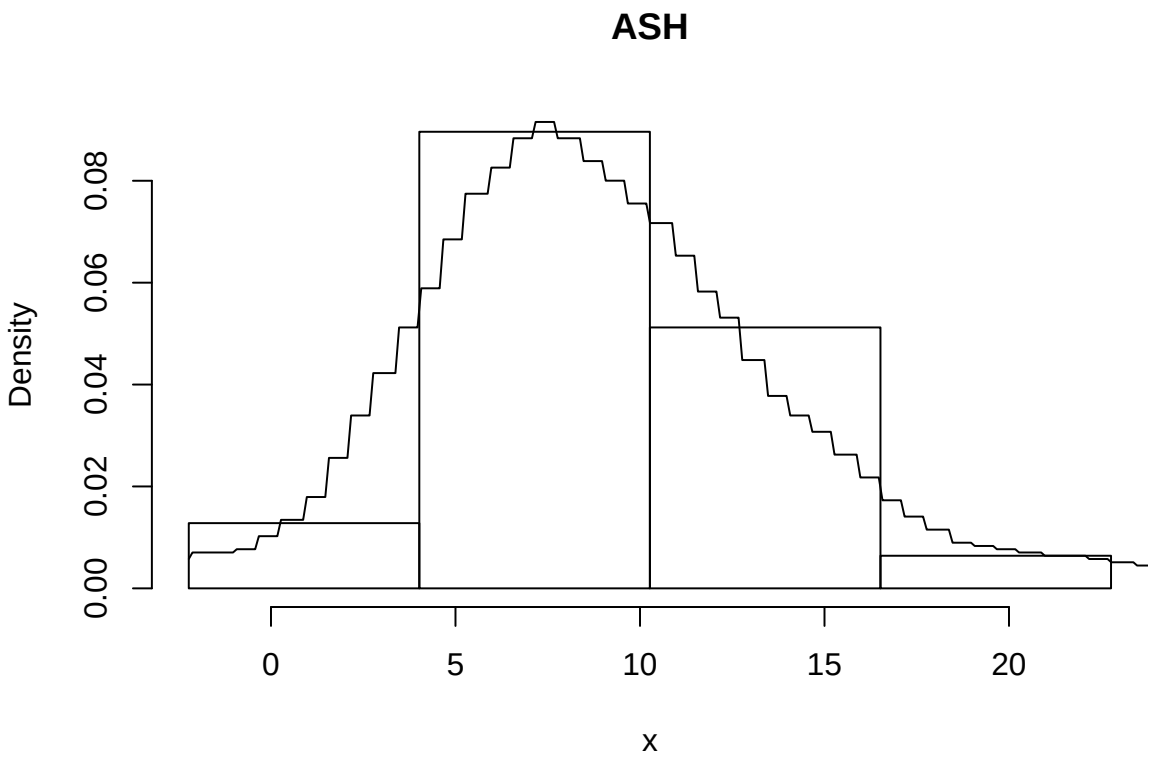
Generate random sample of size n = 25 from Normal(10,25) and using Averaged Shifted Histogram method plot a histogram and comment on the histogram. (use set.seed(1234) to generate same sample)

```r
set.seed(1234)
x <- rnorm(25,10,5)

n <- length(x)
h <- 2.576 * sd(x) * n^(-1/5)

a <- min(x) - .5
b <- max(x) + .5

m <- 10
delta <- h / m

breaks <- seq(a - h, b + 2*h, delta)
hist.ash <- hist(x, breaks = breaks, plot = FALSE)
```

```r
nk <- hist.ash$counts
K <- abs((1-m):(m-1))

fhat <- function(x){
  i <- max(which(x > breaks))
  k <- (i - m + 1):(i + m - 1)
  vk <- nk[k]
  sum((1 - K / m) * vk) / (n * h)
}

fhat(12)
```

```
## [1] 0.05825497
```

```r
# density can be computed at any points in range of data
z <- as.matrix(seq(a, b + h, .1))
f.ash <- apply(z, 1, fhat) #density estimates at midpts


# plot ASH density estimate over histogram
breaks2 <- seq(a, b + h, h)

hist(x, breaks = breaks2, freq = FALSE, main = "ASH", ylim = c(0, max(f.ash)))
lines(z, f.ash, xlab = "x")
```



Kernel Density Estimation:

Kernel density estimation is a non-parametric method of estimating the probability density function (PDF) of a continuous random variable. It is non-parametric because it does not assume any underlying distribution for the variable. Kernel density estimation is a fundamental data smoothing problem where inferences about the population are made, based on a finite data sample.

Let $(x_1, x_2, ..., x_n)$ be a univariate independent and identically distributed sample drawn from some distribution with an unknown density f. We are interested in estimating the shape of this function f. Its kernel density estimator is

$$\hat{f}_K(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{x - X_i}{h})$$

where K is the kernel — a non-negative function — and h > 0 is a smoothing parameter called the bandwidth.

Bandwidth selection:

If Gaussian basis functions are used to approximate univariate data, and the underlying density being estimated is Gaussian, the optimal choice for h (that is, the bandwidth that minimises the mean integrated squared error) is:

$$h = 1.06 * \hat{\sigma} * n^{-1/5}$$

This choice of bandwidth is an optimal (IMSE) choice when the distribution is normal. If the true density is not unimodal, however, will tend to oversmooth. Alternately, one can use a more robust estimate of dispersion in, setting

$$\hat{\sigma} = min(S, IQR/1.34)$$

where S is the standard deviation of the sample. Silverman indicates that an even better choice for a Gaussian kernel is the reduced width

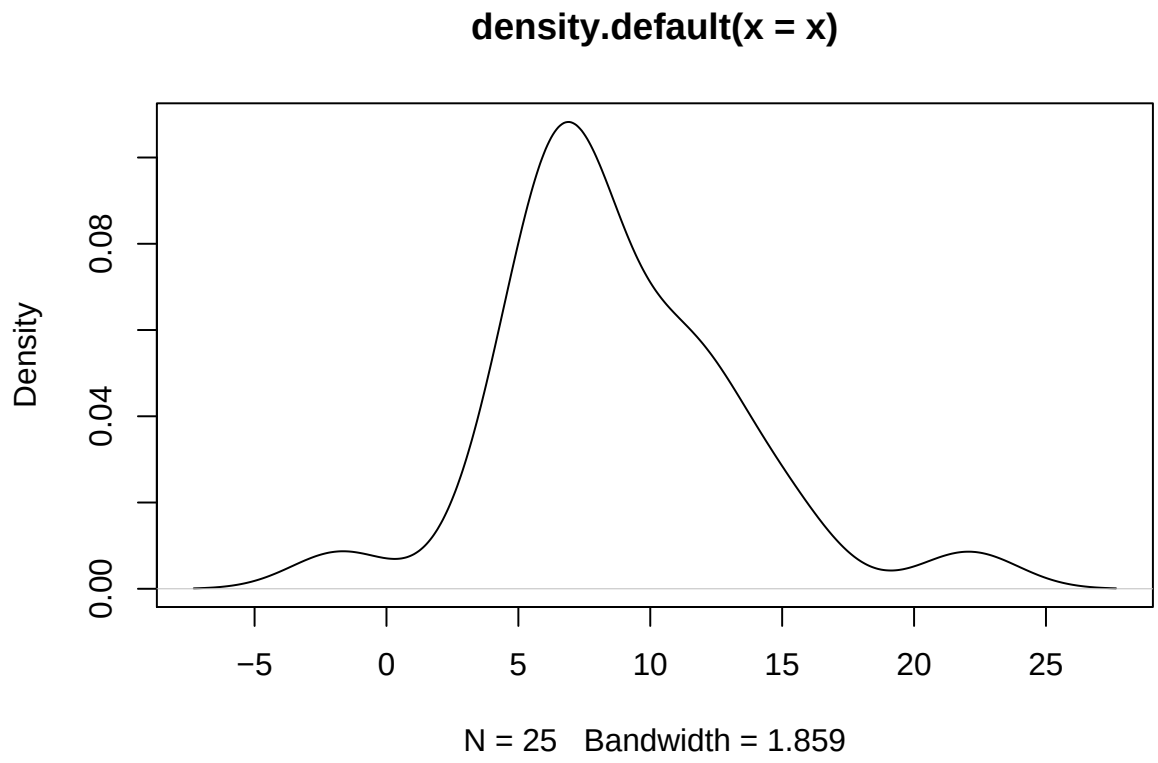
$$h = 0.9 * min(S, IQR/1.34) * n^{-1/5}$$

which is a good starting point appropriate for a wide range of distributions that are not necessarily normal, unimodal, or symmetric.

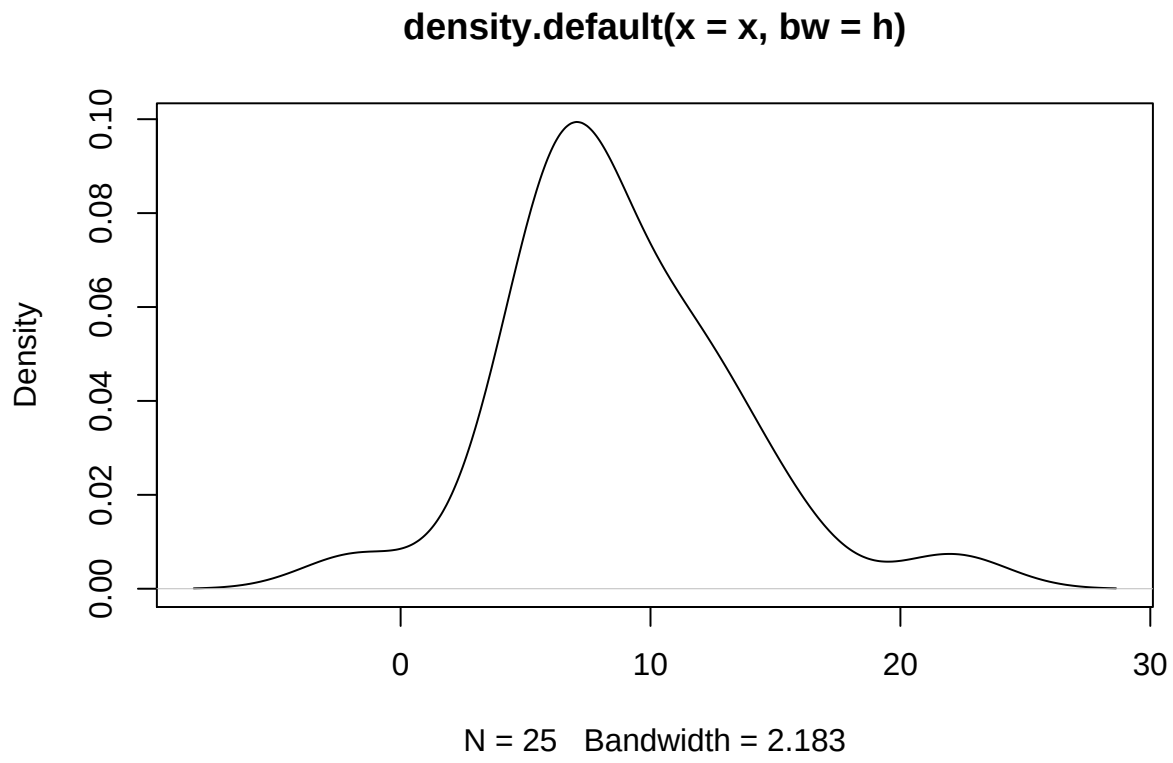
Question 2:

Generate random sample of size n = 25 from Normal(10,25) and using Kernel Density Estimation method (use gaussian kernel) draw density plot and comment on it. (use set.seed(1234) to generate same sample)

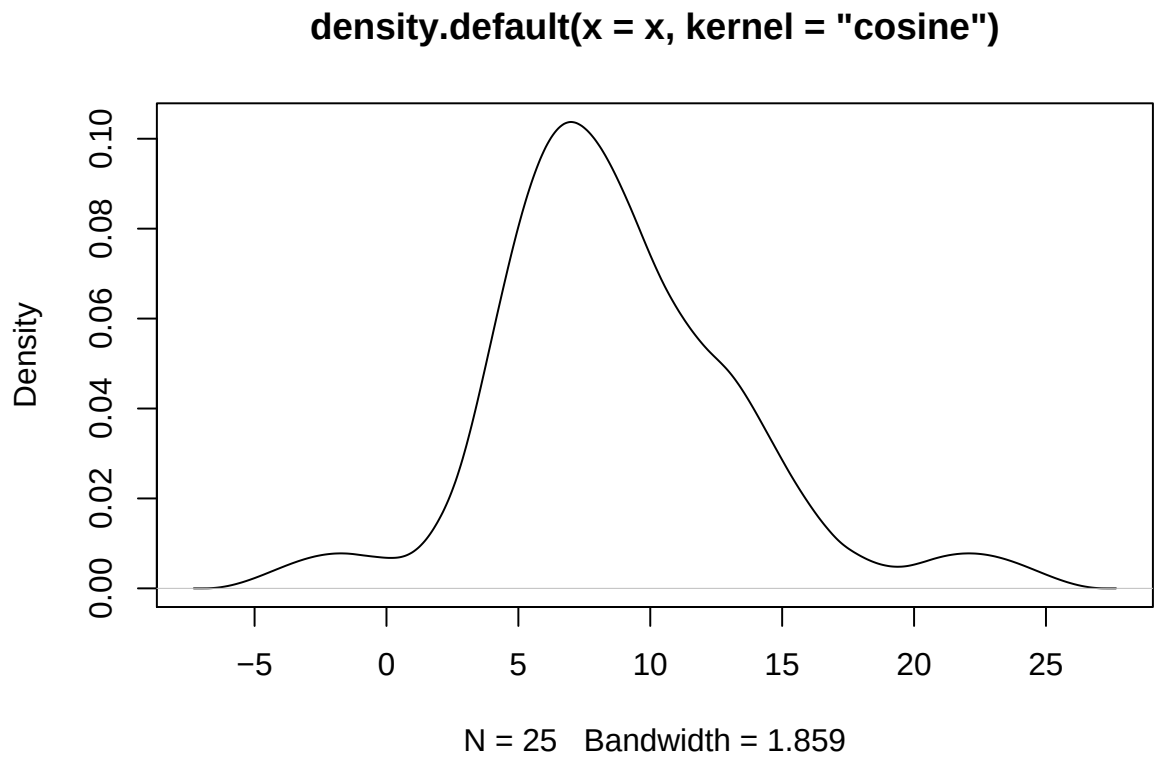
```
set.seed(1234)
x <- rnorm(25, 10, 5)

n <- length(x)
h <- 0.9 * min(sd(x), IQR(x)) * n ^ (-1/5)

plot(density(x))
```



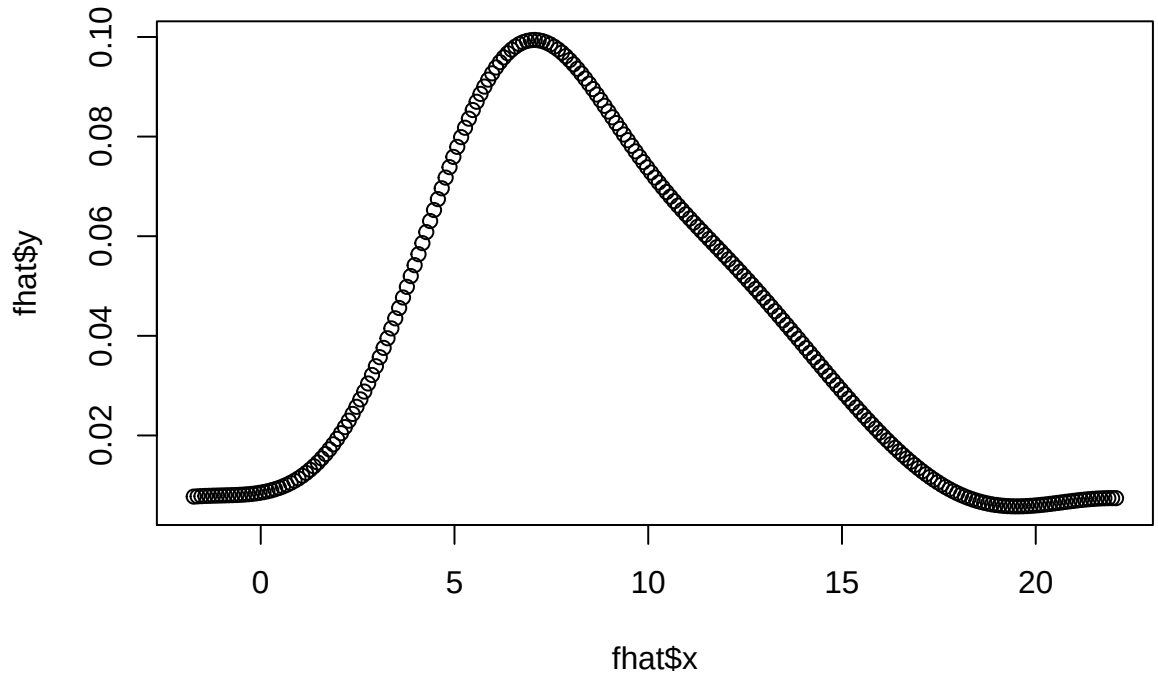

```
plot(density(x, bw = h))
```




```
plot(density(x, kernel = "cosine"))
```




```
d <- density(x, bw = h)
xnew <- seq(min(x), max(x), .1)
fhat <- approx(d$x, d$y, xout = xnew)
plot(fhat)
```




```
approx(d$x, d$y, xout = c(12,21))
```

```
## $x
## [1] 12 21
##
## $y
## [1] 0.05593914 0.00688964
```

```
set.seed(1234)
x <- rnorm(25, 10, 5)

n <- length(x)
h <- 0.9 * min(sd(x), IQR(x)) * n ^ (-1/5)

ND <- function(x){
  sqrt(2*pi)^(-1) * exp(-0.5 * x^2)
}

K <- function(x,data){
  size <- length(x)
  out <- numeric(size)
```

```
  for(i in 1:size){

  out[i] <- mean(ND((x[i]-data)/h)/h)

  }
  return(out)
}

K(c(12,21),x)
```

```
## [1] 0.05588690 0.00688671
```

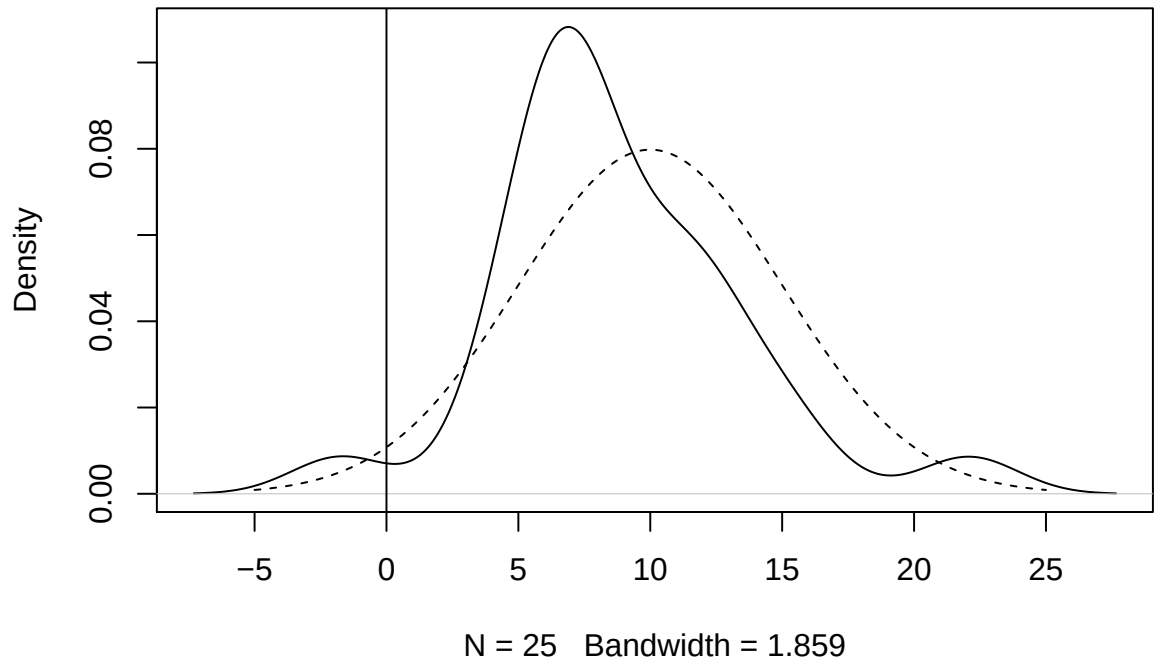
Boundary kernels

Near the boundaries of the support set of a density, or discontinuity points, kernel density estimates have larger errors. Kernel density estimates tend to smooth the probability mass over the discontinuity points or boundary points.

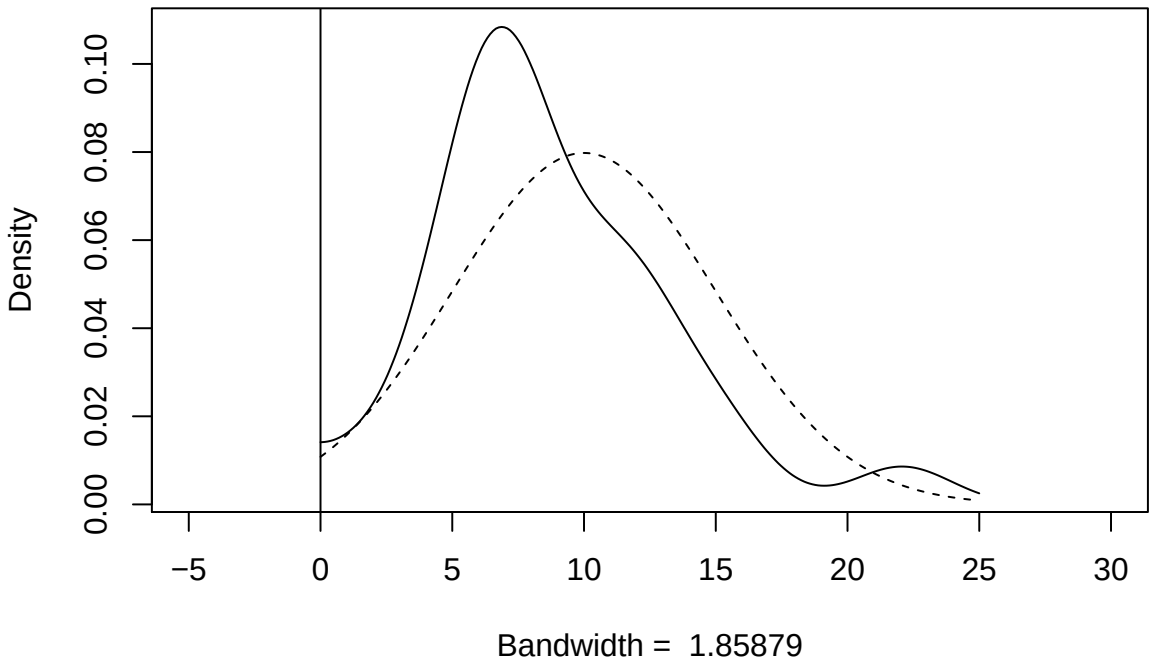
Question 3:

Generate random sample of size n = 25 from Normal(10,25) and draw density plot and then take 0 as boundary and find a new density by using Boundary kernels (use set.seed(1234) to generate same sample)

```
set.seed(1234)
x <- rnorm(25, 10, 5)

n <- length(x)

plot(density(x), main="")
abline(v = 0)

y <- seq(-5,25, .01)
lines(y, dnorm(y, 10, 5), lty = 2)
```




```
#Reflection boundary technique

xx <- c(x, -x)
g <- density(xx, bw = bw.nrd0(x))
a <- seq(0,25, .01)

ghat <- approx(g$x, g$y, xout = a)
fhat <- 2 * ghat$y # density estimate along a

bw <- paste("Bandwidth = ", round(g$bw, 5))

plot(a, fhat, type="l",xlim = c(-5,30), main = "", xlab = bw, ylab = "Density")

abline(v = 0)

# add the true density to compare
y <- seq(0,25, .01)
lines(y, dnorm(y, 10, 5), lty = 2)
```


Density
0.00
0.02
0.04
0.06
0.08
0.10
−5
0
5
10
15
20
25
30
Bandwidth = 1.85879