```
## This R code reads a dataset, constructs and plots Histogram density estimate

dataset = as.matrix(read.table("folderpath/filename.extension", header = T or
 F))

n = length(dataset) ## size of dataset

print("sample size")
print(n)

## specify bin width

h =

## define the bin boundaries

## specify leftmost starting point
## must be smaller than the smallest observation
starting_point =

## end point of the first bin
current_point = starting_point + h

I = c(starting_point,current_point) ## collect the bin boundaries

## does the first bin already cover the entire data ?
## if not, keep creating additional bins

while(current_point <= max(dataset))
{
    ## compute the end point of the next bin
    current_point = current_point + h
    I = c(I,current_point) ## add to existing collection of bin boundaries
}

k = length(I) - 1 ## how many bins needed

## the k bins are:
## [I[1], I[2]), [I[2], I[3]), [I[3], I[4]),...,[I[k], I[k+1])

## calculate how many observations in each bin

count = array(0, length(I) - k)
for (i in 1:k) count[i] = length(which(dataset >= I[i] & dataset < I[i+1]))

## now, determine the histogram density estimate at any point x

fhat_Hn = function(x)
{
    ## density estimate = 0 if point outside the bins
    ## otherwise identify the bin within which x lies
```

```
    if (x < I[1] | x >= I[k+1]) value = 0
    else
    {
    index = which(I > x)[1] - 1 ## which bin x lies inside
    value = count[index]/(n*(I[index+1]-I[index])) ## denominator is (n*bin
     width)
    }
    return(value)
}

## plot Histogram density estimate over the range of values in the dataset

## create a very fine grid of points spanning the range of data
plot_at_points = seq(min(dataset) - 0.1, max(dataset)+ 0.1, by = 0.0005)

## now calculate histogram density estimate at chosen points

fhat_Hn_values = sapply(plot_at_points,fhat_Hn)

## Now create the plot
x11()

plot(plot_at_points,fhat_Hn_values,"l",ylab="",main="")
```