```
## This R code reads a dataset, constructs and plots Kernel density estimate
## allows to choose from uniform, triangular, quadratic and Gaussian kernels

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

n = length(dataset) ## size of dataset

print("sample size")
print(n)

## specify bandwidth

h =

## define kernel functions that we learnt in class

## the ifelse function is checking for neighbors
## needed for uniform, triangular and quadratic kernels
## not needed for Gaussian kernel

## x is the point at which we calculate the density estimate
## x_i is the placeholder for observations in tthe dataset

uniform_kernel_function = function(x,x_i,h){ifelse(abs(x-x_i)< h, 1/(2*h), 0)}

triangular_kernel_function = function(x,x_i,h){ifelse(abs(x-x_i)< h, 1/h -
 abs(x-x_i)/(h*h), 0)}

quadratic_kernel_function = function(x,x_i,h){ifelse(abs(x-x_i)< h, 3/(4*h) -
 (3/4)*((x-x_i)^2)/(h^3), 0)}

Gaussian_kernel_function =
 function(x,x_i,h){(1/sqrt(2*pi*h*h))*exp(-0.5*((x-x_i)^2)/(h*h))}

## now calcuulate the kernel density estimate at x

fhat_kn = function(x,kernel_function)
{
    value = 0
    ## evaluate the kernel function for each observation in the data and sum
    for (i in 1:n) value = value + kernel_function(x,dataset[i],h)
    ## divide by the number of observations in the dataset
    value = value/n
    return(value)
}

## plot the kernel density estimate as a curve 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, specifying which
 kernel you want to use

fhat_kn_values = sapply(plot_at_points,fhat_kn, kernel_function = Write the
 name of chosen kernel function here)

## Now create the plot
x11()

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