Statistics - Week 3

Jimmy Tsz Ming Yue*

University of Sydney

jyue6728@uni.sydney.edu.au

Semester 2 Statistics 2018

Contents

1 Introduction to Density Estimation

In this section of the unit we will introduce density estimation and smoothing methods. There are two types of methods classified as *parametric nonparametric*. Compared to *parametric* methods *nonparametric* methods lack of a formal statistical model. These methods are generally intended for description rather than formal inference. We learn such methods to describe the probability distribution of univariate random variable and multivariate random variables.

*440159151

1.1 Usage of estimating a density function

In exploratory data analysis, an estimate of the density function can be used:

1. to assess multimodality, skew, tail behaviour, etc
2. in decision making, classifcation and summarising Bayesian posteriors.
3. as a useful visulaisation tool.

In this section, the concern is the estimation of a density function f using observations of random variables $x_1, \dots x_n$ sampled independently from f.

2 Parametric density estimation

The parametric approach to a density estimation assumes a parametric model, $x_1, \dots x_n \sim$ i.i.df_θ (where θ is a low-dimensional parameter vector).

For example let us assume $x_i \sim N(m, s)$. We typically estimate parameters $\hat{\theta}$ using maximum likelihood. From this, density at x can be estimated as $f(x|\hat{\theta})$.

Definition. Likelihood The number that is the probability of some observed outcomes given a set of parameter values is regarded as the lieklihood of the set of parameter values given the observed outcomes. To put this in terms of probability densities; let us denote the probability density function assosciated with the outcomes O as $f(O|\theta)$. Then we can estimate θ given observed outcomes O by maximising the function:

$$L(\theta|O) = f(O|\theta) \tag{1}$$

Then let $f(x_1, x_2 \dots, x_n|\theta)$ be the probability of observing $x_1 \dots x_n$ given parameters θ. (It should be a constant reminder that θ is the model and that x_i are data. From this we have that:

$$\begin{aligned} f(x_1, \dots x_n|\theta) &= f(x_1|\theta) \cdot f(x_2|\theta) \cdot \dots \cdot f(x_n|\theta) \\ &= \prod_{i=1}^{n} f(x_i|\theta) \end{aligned}$$

Then the likelihood function is defined as:

$$L(\theta|x_i) = \prod_{i=1}^{n} f(x_i|\theta)$$

as it is often easier to maximise log likelihoods we then have using log laws;

$$\ln L(\theta|x_i) = \sum_{i=1}^{n} \ln f(x_i|\theta)$$

3 Non-Parametric Density Estimation

There are some dangers when using the parametric approaches that one should be careful of; When the assumed model f_θ is incorrect this approach can lead to serious inferential errors. We can use nonparametric approaches to density estimation, which:

1. Assume very little about the form of f.

2. use local information to estimate f at a point x.

An example of a nonparametric density estimator are Histograms which are a certain type of nonparametric density estimator, being piecewise constant density. Most mathematically packages support automatic histogram generation.

3.1 Histogram

A histogram can be used as a simple visualisation of data. In a histogram, bins are first defined and then the number of data points within each bin is used to determine the height of the bar plot. With differing bin sizes histograms may look different.

4 Demonstrate histogram and parametric methods

We can smooth out histograms using a smoother kernel, called a gaussian kernel density estimate. Each point contributes to a Gaussian curve to the total. This kernel is:

1. A smooth density estimate derived from the data

2. A powerful non-parametric model of the distribution of points.

4.1 Kernel

Definition. A kernel is a special type of probability density function which is symmetric. A kernel is a function and has the following properties:

1. Non-negative

2. real-valued

3. symmetric

4. Its definite integral over its support set must equal to 1.

4.1.1 Kernel Density Estimation

Kernel Density Estimation is a non-parametric approach for 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. (X does not need to be assumed to follow any specific distribution.) At every point, a kernel function is created with the point at its centre. The kernel is symmetric around the point. We can estimate the probability density function is estimated by adding all of these kernel functions and dividing by the number of data to ensure that it satisfies the following:

1. every possible value of the pdf is non-negative

2. the definite integral of the pdf over its support set equals 1

The simple density estimator that weights all points within h of x equally is given by:

$$\hat{f}(x) = \frac{1}{2hn} \sum_{i=1}^{n} 1_{\{x - Xi < h\}} \tag{2}$$

A univariate kernel density estimator allows a more flexible weighting scheme, fitting

$$\hat{f}(x) = \frac{1}{hn} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{3}$$

where K is a kernel function and h is a fixed number, called the bandwidth, window width or smoothing parameter.

4.1.2 Constructing a kernel density estimator

1. Choose a kernel; the common ones are normal (Gaussian), uniform (rectangular) and triangular. (Kernel functions are positive everywhere and symmetric at zero).
2. At each point X_p build the scaled kernel function

$$\frac{1}{h} K\left[\frac{(x - X_i)}{h}\right] \tag{4}$$

   where K is a kernel function and h is a fixed number, called the bandwidth, window width or smoothing parameter.
3. Add the individual scaled kernel functions and divide by n;

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left[\frac{(x - X_i)}{h}\right] \tag{5}$$

   this places a probability of $1/n$ to each x_i, which ensures the kernel density estimate integrates to 1.

Intuitvely we can think of distributions as the sum of many bumps.

4.1.3 Choice of Kernels

Kernel density estimation requires specifcation of two components; the kernel and the bandwith.

The shape of the kernel has much influence on the results than does the bandwidth. We have the following examples;

1. Uniform:

$$k_0(u) = \frac{1}{2}(|u| \leq 1) \tag{6}$$

2. Epanechnikov:

$$k_1(u) = \frac{3}{4}(1 - u^2)(|u| \leq 1) \tag{7}$$

3. Biweight:

$$k_2(u) = \frac{15}{16}(1 - u^2)^2(|u| \leq 1) \tag{8}$$

4. Triweight:

$$k_3(u) = \frac{35}{32}(1 - u^2)^3(|u| \leq 1) \tag{9}$$

5. Triweight:

$$k_\phi(u) = \frac{1}{\sqrt{2\pi}}\left(-\frac{u^2}{2}\right) \tag{10}$$

5 Demonstrate Kernel Density Estimation

The density estimator:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left[\frac{(x - X_i)}{h} \right] \tag{11}$$

is a fixed-bandwidth kernel density estimator since h is constant. The value chosen for the bandwith exerts a strong influence on the estimator $\hat{f}$.

1. If h is too small, the density estimator will tend to assign probability density too locally near observed data.
2. If h is too large, the density estimator will spread probability density contributions too diffusely.

The bandwidth determines the trade-off between the bias and variance of estimator $\hat{f}$.

1. A small bandwidth produces a density estimator with function indivative of high variability caused by under smoothing.
2. A large bandwidth causes important features of f to be smoothed away, thereby causing bias.

The trade-off between the bias and the variane of an estimator is important in nearly all kinds of model selection.

5.1 Integrated Squared Error

To evaluate $\hat{f}$ as an estimator of f over the entire range of support, one could use the integrated squared error defined as:

Definition. ISE

$$\text{ISE}(h) = \int_{-\infty}^{\infty} \left[\hat{f}(x) - f(x) \right]^2 dx \tag{12}$$

measuring the square of distances between $\hat{f}(x)$ to $f(x)$

To discuss the generic properties of an estimator, without reference to a particular observed sample, it is more sensible to average ISE(h) over all samples that might be oberved for which we can define:

Definition. Mean Integrated Squared Error(MISE)

$$\text{MISE}(h) = E\left\{ \text{ISE}(h) \right\} \tag{13}$$

6 Cross-Validation

Cross-Validation, is a model validation for assessing how the results of a statistical analysis will generalise to an independent data set. We use this mainly when one wants to estimate how accurately a predicitve model will perform in practice. In a prediction problem, a model is usually given and we have a:

1. a dataset of known data on which training is run (training dataset)
2. datset of unknown data against which the model is tested.

6.1 Goal of cross-validation

The goal of cross validation is to define a dataset to “test” the model in the training phase (validation dataset) in order to:

1. limit problems like overfitting
2. Give insight on how the model will generalise to an independent dataset.

One round of cross-validation invovles:

1. Partitiong a sample of data into complementary subsets,
2. performing the analysis on one subset (called the training set) and
3. validating the analysis on the other subset (called the vaildation or the testing set)

6.2 Why use Cross-Validation

Many bandwidth selection strategies begin by relating h to some measure of the quality of $\hat{f}$ as an estimator of f. The quality is quantified by some $Q(h)$, whose estimate $\hat{Q}(h)$, is optimised to find h. If $\hat{Q}(h)$ evaluates the quality of $\hat{f}$ based on how well it fits the observed data, this implies that observed data are being used twice:

1. Once to calculate $\hat{f}$ from the data and a
2. second time to evaluate the quality of $\hat{f}$ as an estimator of f

This double use prvodies an overoptimistif view of the quality of the estimator leading to overfitting. If we use cross-validation this problem is voided.

7 Spline Methods

7.1 Cubic Spline

A cubic spline is a piecewise cubic function that is everywhere twice differentiable.

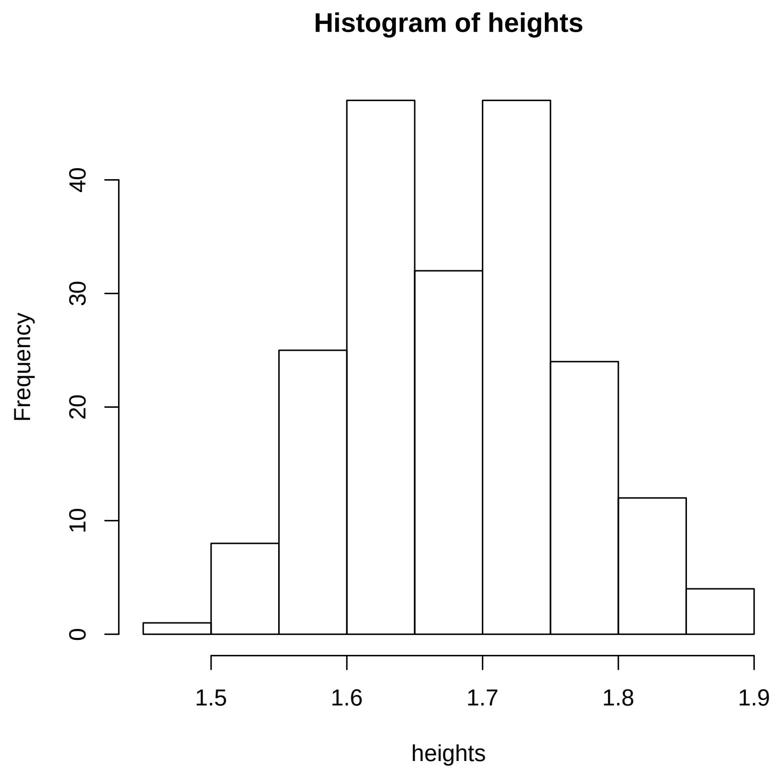
8 Tutorial

The height.txt file (please download from week 3, Datasets Zip link and unzip the file) contains heights of 100 men and 100 women measured in meters. There is another file called newHeight.txt which contains additional heights of 10 men and 10 women.

1. Create an R markdown file, read in height.txt data and generate histogram to summarise and visualise height variable.

Solution. We first read in the height file

```
> height <- read.delim("height.txt", header=TRUE)
> #head(height)
> name <- c("iD", "Height(m)")
> df <- data.frame(height)
> colnames(df) <- name
> heights <- as.numeric(df$"Height(m)")
> hist(heights)
```

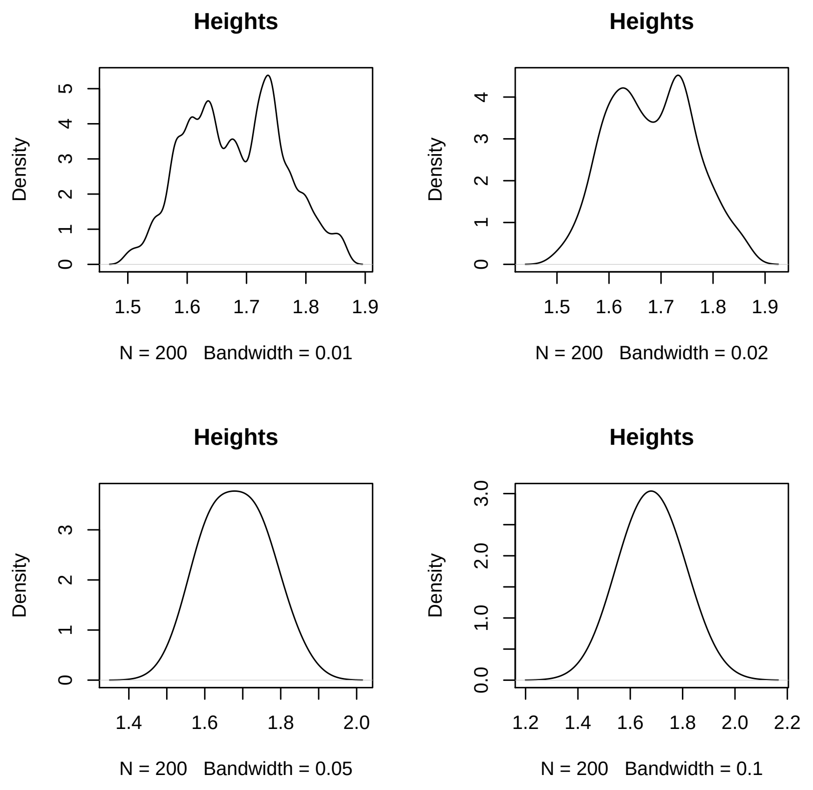


□

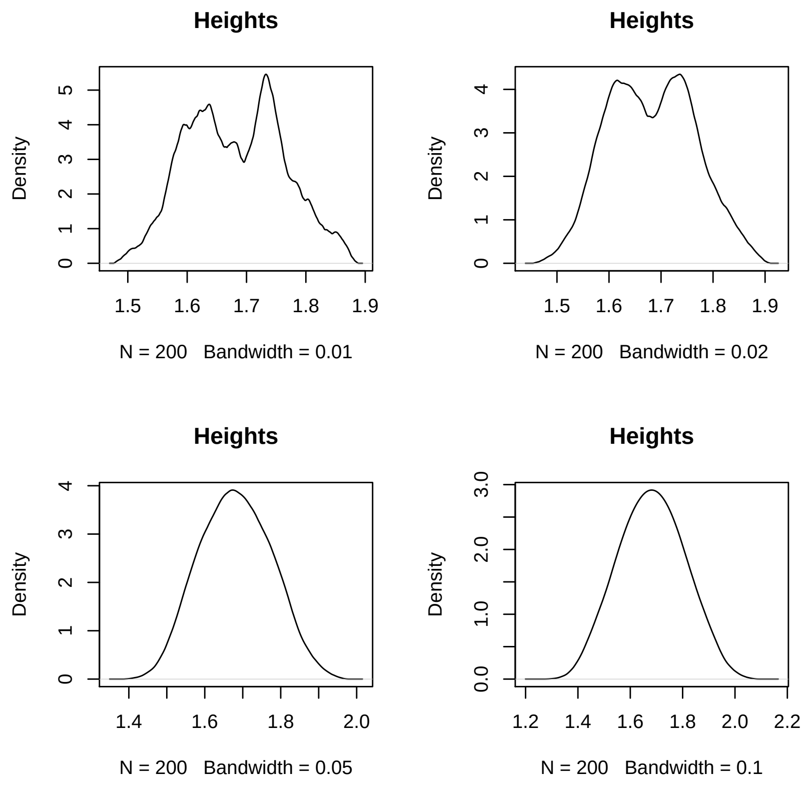
2. Apply kernel density estimation methods to estimate height density. Try different kernel functions and bandwidths.

Solution. We use the density function in R:

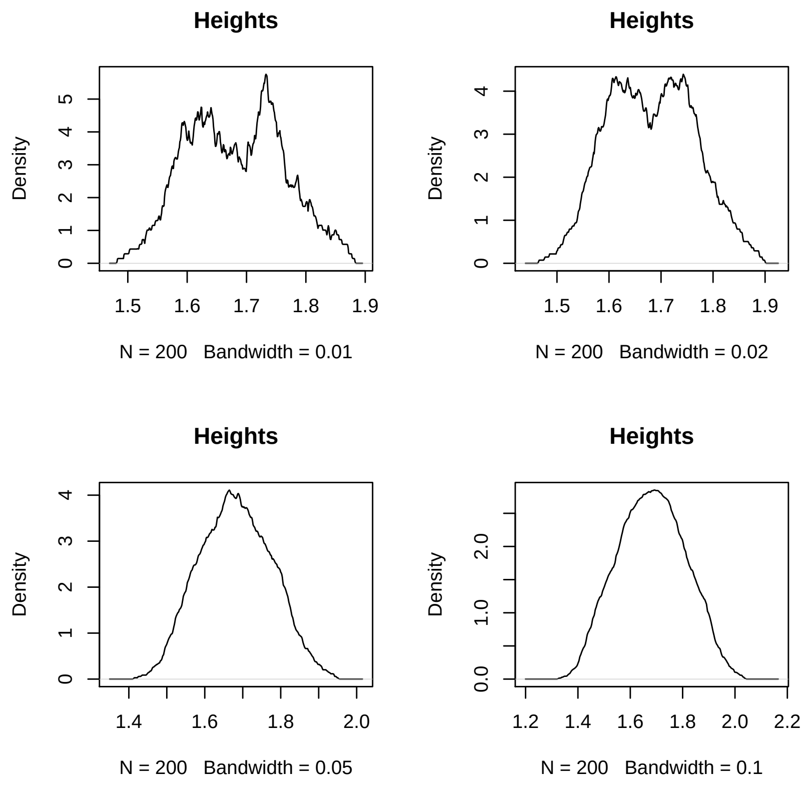
```
> kernel = c("gaussian", "epanechnikov", "rectangular",
+                     "triangular", "biweight",
+                     "cosine", "optcosine")
> d1 <- density(heights, window = "gaussian", bw = 0.01)
> d2 <- density(heights, window = "gaussian", bw = 0.02)
> d3 <- density(heights, window = "gaussian", bw = 0.05)
> d4 <- density(heights, window = "gaussian", bw = 0.10)

> par(mfrow = c(2,2))
> plot(d1, main = "Heights")
> plot(d2, main = "Heights")
> plot(d3, main = "Heights")
> plot(d4, main = "Heights")
```



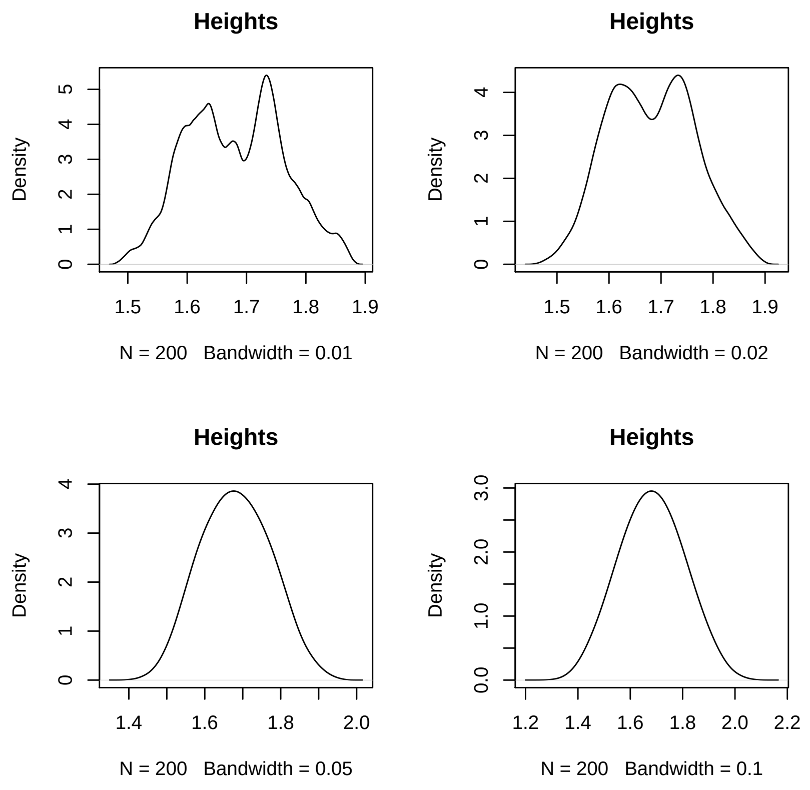
```
> kernel = c("gaussian", "epanechnikov", "rectangular",
+                     "triangular", "biweight",
+                     "cosine", "optcosine")
> d1 <- density(heights, window = "epanechnikov", bw = 0.01)
> d2 <- density(heights, window = "epanechnikov", bw = 0.02)
> d3 <- density(heights, window = "epanechnikov", bw = 0.05)
> d4 <- density(heights, window = "epanechnikov", bw = 0.10)

> par(mfrow = c(2,2))
> plot(d1, main = "Heights")
> plot(d2, main = "Heights")
> plot(d3, main = "Heights")
> plot(d4, main = "Heights")
```



```
> kernel = c("gaussian", "epanechnikov", "rectangular",
+                     "triangular", "biweight",
+                     "cosine", "optcosine")
> d1 <- density(heights, window = "rectangular", bw = 0.01)
> d2 <- density(heights, window = "rectangular", bw = 0.02)
> d3 <- density(heights, window = "rectangular", bw = 0.05)
> d4 <- density(heights, window = "rectangular", bw = 0.10)

> par(mfrow = c(2,2))
> plot(d1, main = "Heights")
> plot(d2, main = "Heights")
> plot(d3, main = "Heights")
> plot(d4, main = "Heights")
```



```
> kernel = c("gaussian", "epanechnikov", "rectangular",
+                     "triangular", "biweight",
+                     "cosine", "optcosine")
> d1 <- density(heights, window = "biweight", bw = 0.01)
> d2 <- density(heights, window = "biweight", bw = 0.02)
> d3 <- density(heights, window = "biweight", bw = 0.05)
> d4 <- density(heights, window = "biweight", bw = 0.10)

> par(mfrow = c(2,2))
> plot(d1, main = "Heights")
> plot(d2, main = "Heights")
> plot(d3, main = "Heights")
> plot(d4, main = "Heights")
```



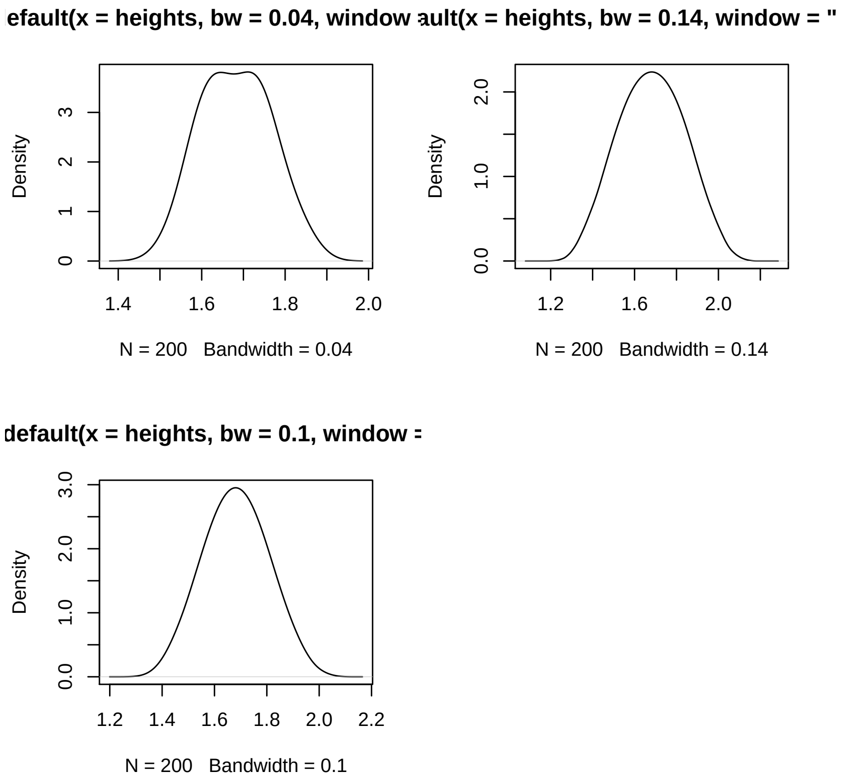
□

3. Use BCV methods to select for optimal bandwidth for each kernel of choice.

Solution. we use the bcv function in R:

```
> obw <- bw.bcv(heights)
> obw

[1] 0.03196391

> library(kedd)
> obw.gauss <- h.bcv(heights, kernel = "gaussian")
> obw.epan <- h.bcv(heights, kernel = "epanechnikov")
> obw.rect <- h.bcv(heights, kernel = "biweight")
> obw.rect

Call:                Biased Cross-Validation 1

Derivative order = 0
Data: heights (200 obs.);        Kernel: biweight
Min BCV = 0.05189653;        Bandwidth 'h' = 0.107642

> obw.epan
```

```
Call:                Biased Cross-Validation 1

Derivative order = 0
Data: heights (200 obs.);        Kernel: epanechnikov
Min BCV = 0.2345472;        Bandwidth 'h' = 0.1418205

> obw.gauss

Call:                Biased Cross-Validation 1

Derivative order = 0
Data: heights (200 obs.);        Kernel: gaussian
Min BCV = 0.05893172;        Bandwidth 'h' = 0.04177944

> par(mfrow=c(2,2))
> plot(density(heights, window = "gaussian", bw =0.04))
> plot(density(heights, window = "epanechnikov", bw =0.14))
> plot(density(heights, window = "biweight", bw =0.10))
```

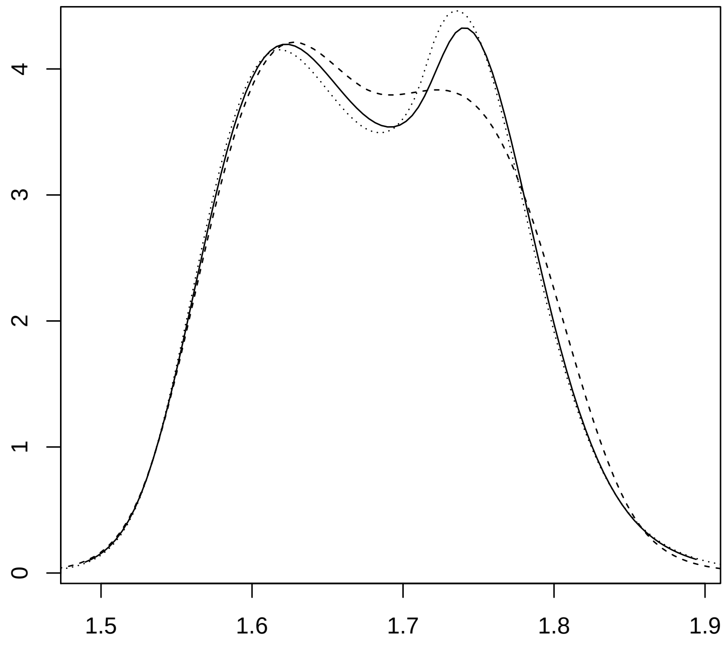


□

4. Apply cubic spline density estimation with different number of knots and compare these results to those from using kernel functions.

Solution. We fist import the package and follow the given example

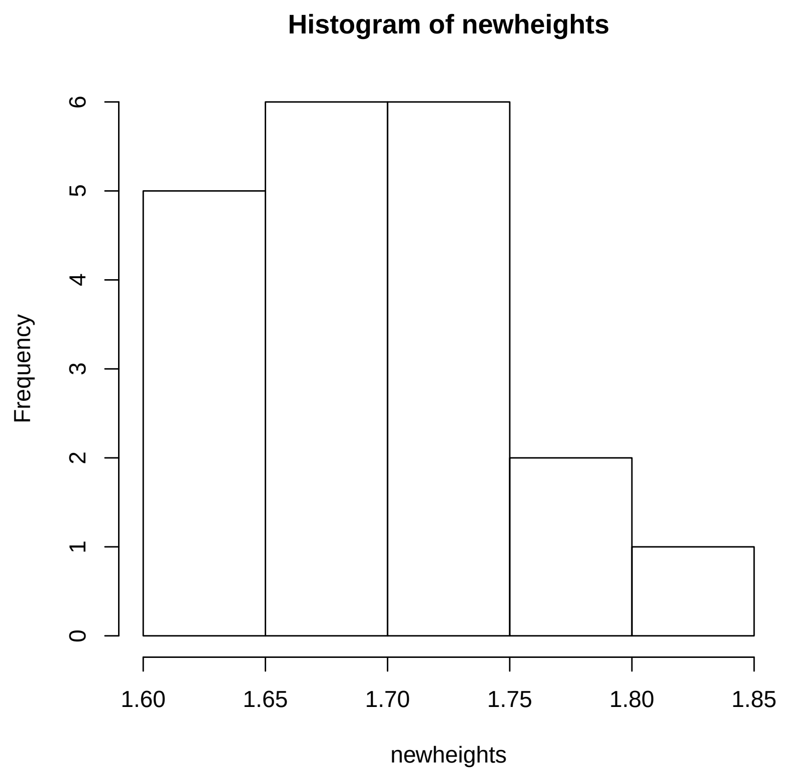
```
> library(polspline)
> fit1 <- logspline(heights)
> fit2 <- logspline(heights, nknots=5)
> fit3 <- logspline(heights, nknots=10)

> plot(fit1)
> plot(fit2, add=T,lty=2)
> plot(fit3, add=T,lty=3)
>
>
```



□

5. Read in newHight.txt data. Is it possible to classify each of these new samples in newHight.txt dataset based on density estimation results from height.txt? Try to perform such classification.

Solution.
```
newheight <- read.delim("newHeight.txt", header=TRUE)
> #head(height)
> name <- c("iD", "Height(m)")
> df <- data.frame(newheight)
> colnames(df) <- name
> newheights <- as.numeric(df$"Height(m)")
> hist(newheights)
```



We may make the estimation that the newheights may be drawn from the original distribution. However more validation on this is necessary □