Lab Week 3

STAT5003

Dr. Justin Wishart

Semester 2, 2022

Contents

Preparation and assumed knowledge

- Density estimation content covered in Module 3.

Aims

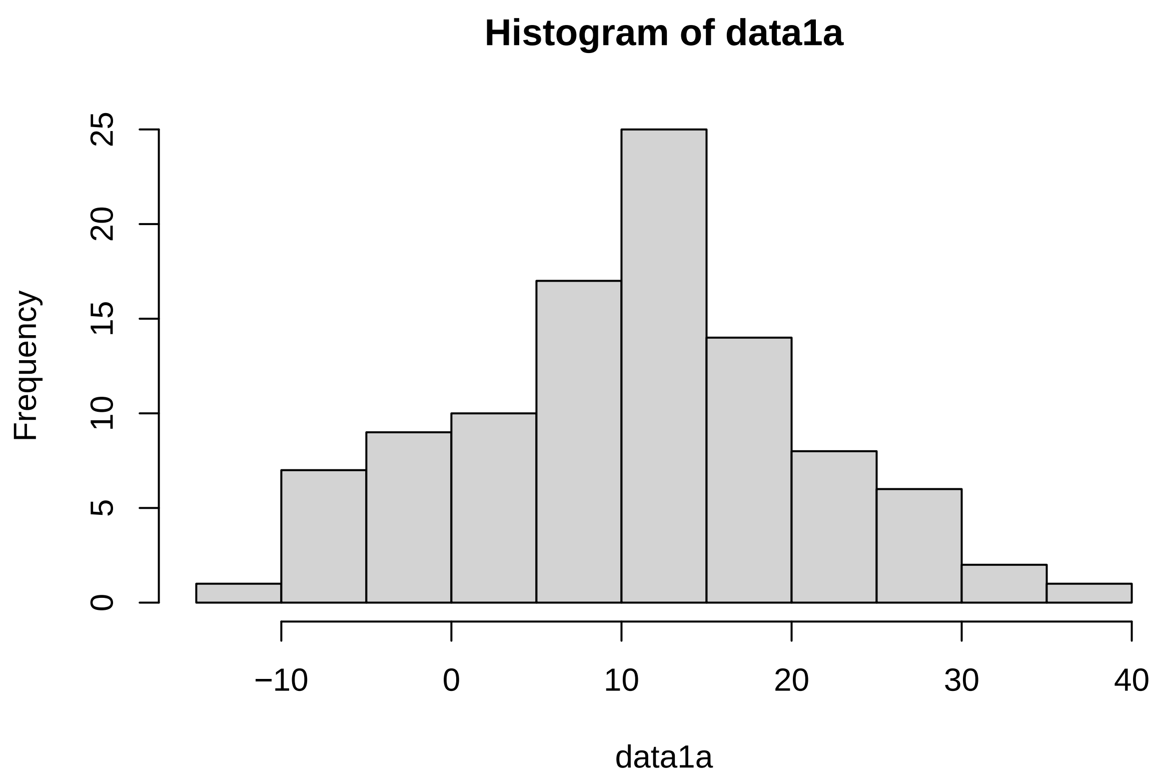
- Learn to simulate data from probability distributions.
- Consider the MLE estimators.
- Consider Nonparametric kernel smoothing estimators.
- Learn about reproducibility with `set.seed`.

1 Parametric density estimation

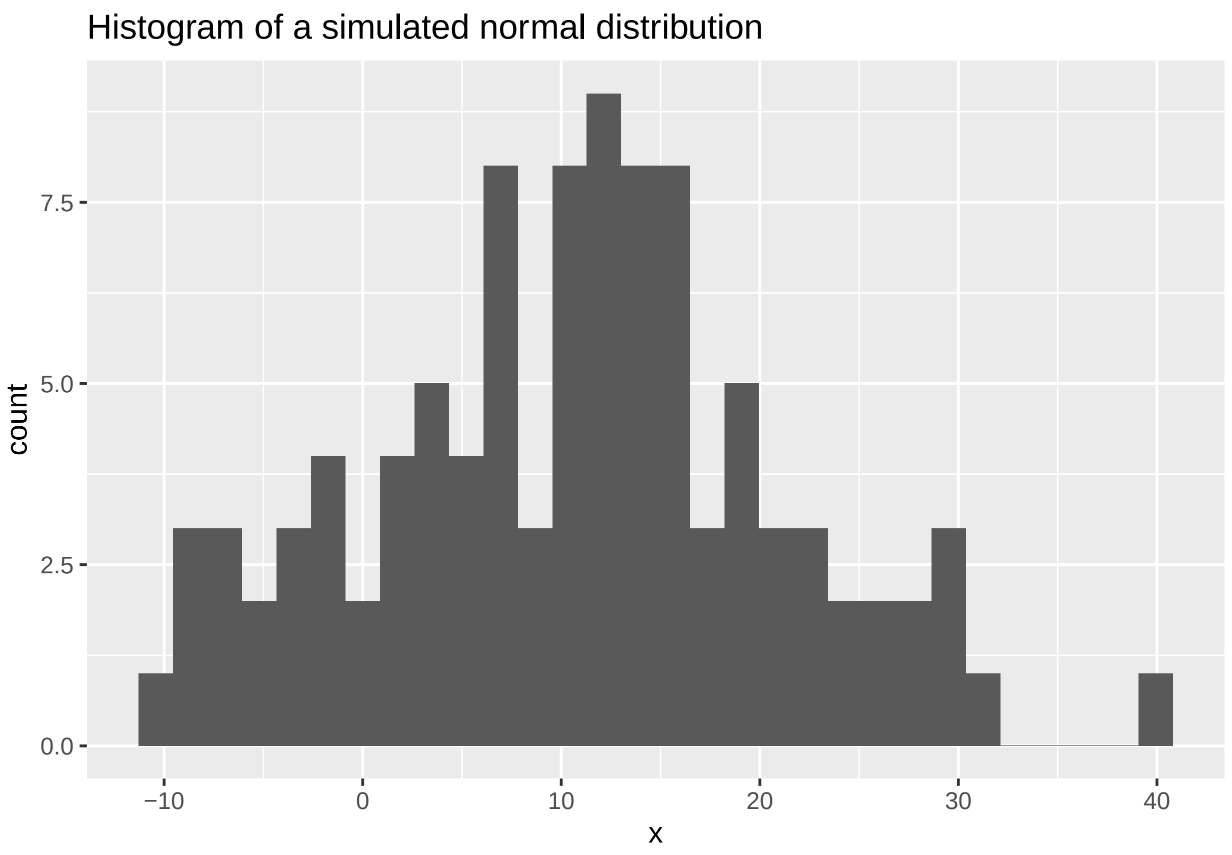
1.1 Basic Gaussian simulation

Simulate 100 samples of a normal distribution (`rnorm`) with mean of 10 and standard deviation of 10. Plot a histogram of this sample.

```r
data1a <- rnorm(100, mean = 10, sd = 10)
hist(data1a)
```



```
## or ggplot way
library(ggplot2)
ggplot(data.frame(x = data1a), aes(x = x)) + geom_histogram(bins = 30) +
    ggtitle("Histogram of a simulated normal distribution")
```



1.2 Reproducible RNG

Set the random seed to your student number using `set.seed`. Then create another sample from a normal distribution with a mean of 10 and sd of 10 and 100 observations. Demonstrate that if the seed is reset again to your student number, you can generate another 'random' sample that gives the same output. (Hint: the functions `identical` or `all.equal` are useful for this. The difference between these functions is discussed here.)

```
set.seed(5003)
x <- rnorm(100, mean = 10, sd = 10)
y <- rnorm(100, mean = 10, sd = 10)
set.seed(5003)
z <- rnorm(100, mean = 10, sd = 10)
identical(x, y)
```

```
## [1] FALSE
```

```
identical(x, z)
```

```
## [1] TRUE
```

1.3 Log likelihood

Simulate another 100 samples of a normal distribution, this time with mean of 8 and standard deviation of 10. What is the log likelihood that this set of 100 observations are drawn from a normal distribution parameterised as $\mathcal{N}(\mu = 10, \sigma = 10)$? Calculate the log likelihood for a range of parameter values $\mu = 1, 2, ...20$, keeping $\sigma = 10$. Plot the log likelihood against μ. From the plot, what is the maximum likelihood estimate of μ?

The log likelihood is as follows

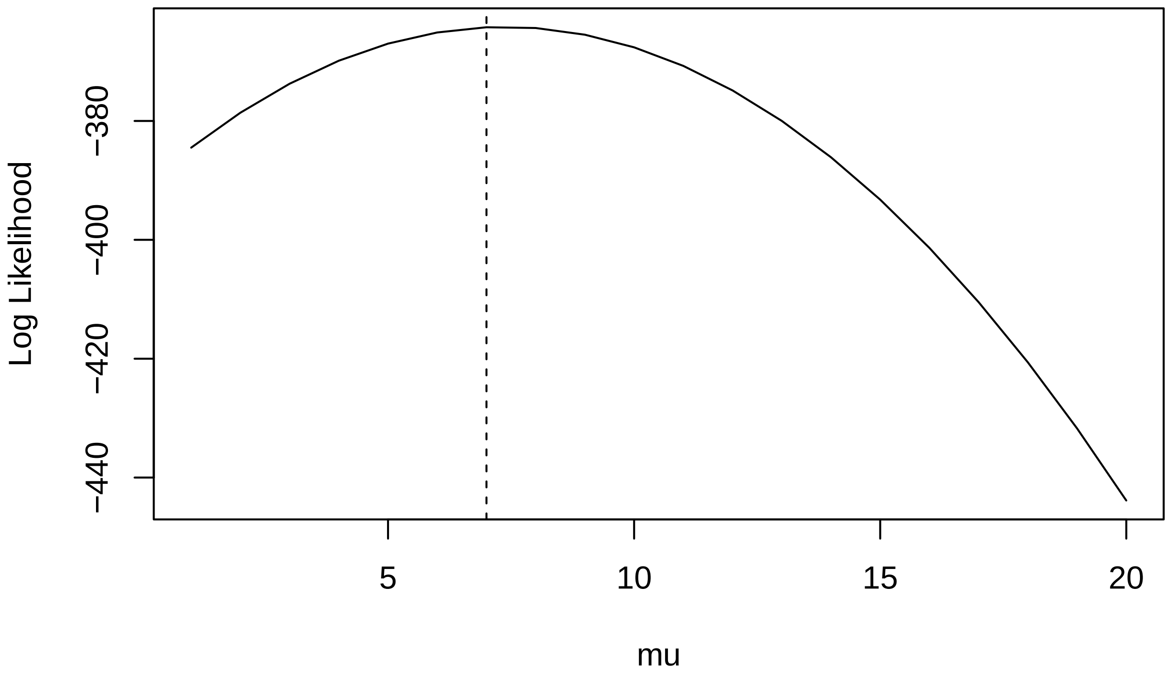
$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{x}) = \log \prod_{i=1}^{n} \left(\sqrt{2\pi\sigma^2}\right)^{-1} \exp\left\{-\frac{(x_i - \mu)^2}{2\sigma^2}\right\} = -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

You can do this question by defining the likelihood used above as a function or transforming the Gaussian density in R directly.

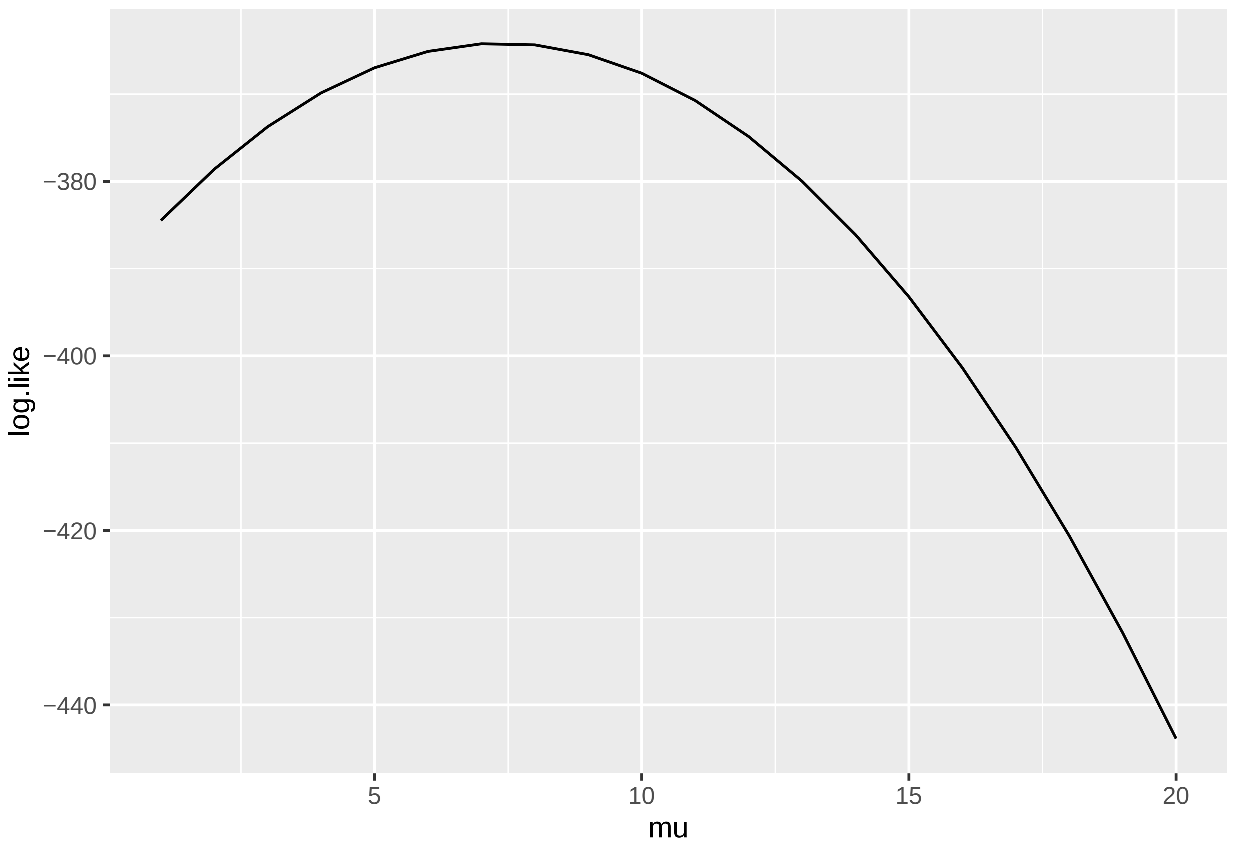
```r
data.1c <- rnorm(100, 8, 10)
mu <- 1:20
log.likelihood <- sapply(mu, function(x) sum(log(dnorm(data.1c, mean = x, sd = 10))))
plot(mu, log.likelihood, type = "l", ylab = "Log Likelihood")
mu.mle <- mu[which.max(log.likelihood)]
mu.mle
```

```
## [1] 7
```

```r
abline(v = mu.mle, lty = 2)
```



```r
# alternatively with ggplot/tidyverse
likelihood.tbl <- tibble(mu) |>
    mutate(log.like = purrr::map_dbl(mu, \(x) sum(log(dnorm(data.1c, mean = x, sd = 10)))))
ggplot(likelihood.tbl, aes(x = mu, y = log.like)) +
    geom_line()
```

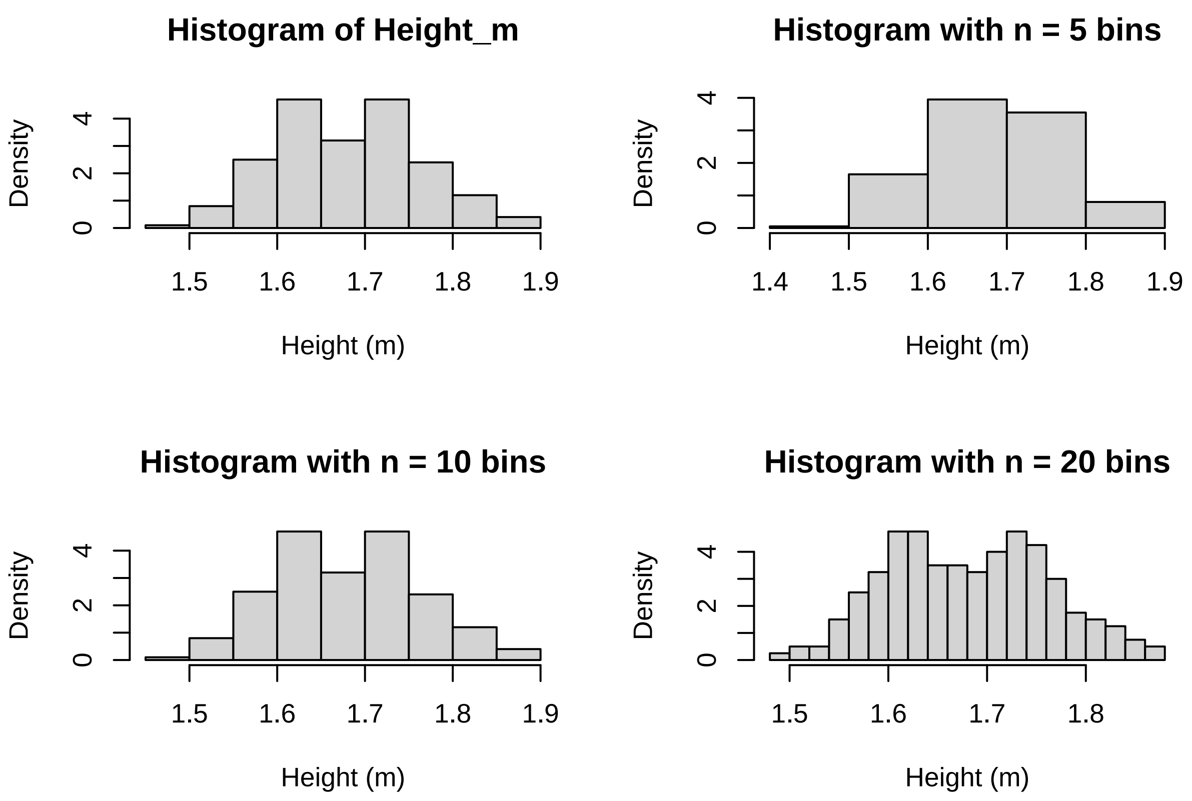


2 Parametrics and Nonparametric density estimation

2.1 Basic exploratory data analysis

For this question, we will be performing density estimates on the heights dataset. Load the heights data into a data frame (available from Canvas). Make a histogram of the heights, set the y axis to be probability not frequency. Repeat for 5, 10 and 20 bins.

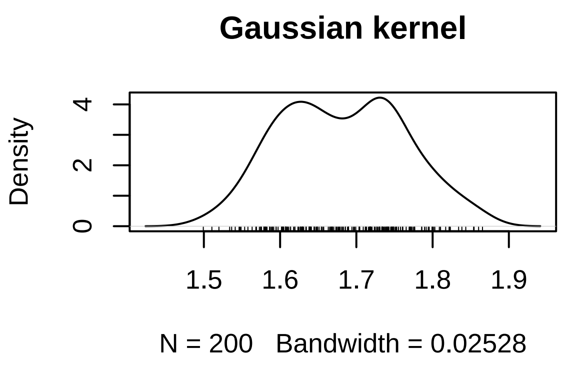
```
heights <- read.table("height.txt", header = TRUE)
# Create 2x2 plotting grid
par(mfrow = c(2, 2))
with(heights, hist(Height_m, probability = TRUE, xlab = "Height (m)"))
n.bins <- c(5, 10, 20)
invisible(lapply(n.bins, function(n) {
  hist(heights[["Height_m"]], probability = TRUE, breaks = n,
       xlab = "Height (m)",
       main = paste0("Histogram with n = ", n, " bins"))
}))
```

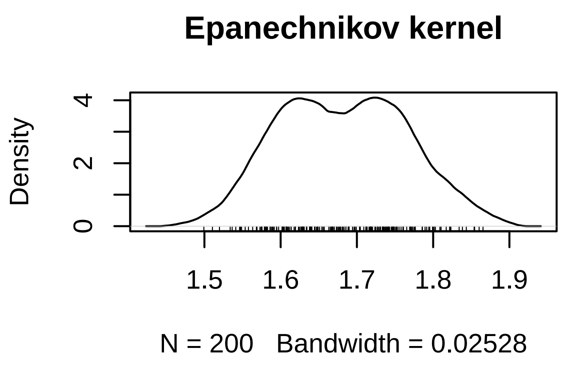


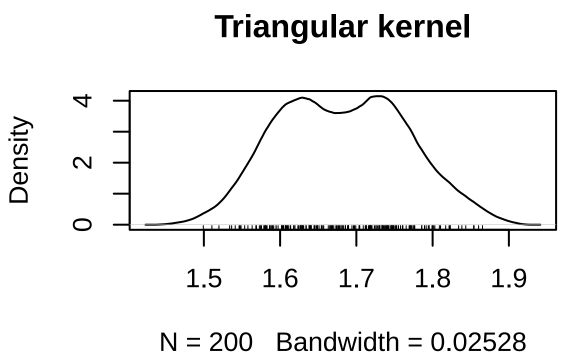
2.2 Kernel density estimation

Use the `density` function to perform kernel density estimation. Do this with the Gaussian, the Epanechnikov and the triangular kernels. Plot the results.

```r
kernels <- c("gaussian", "epanechnikov", "triangular")
par(mfrow = c(2, 2))
invisible(lapply(kernels, function(k) {
    plot(density(heights$Height_m, kernel = k),
         main = paste(tools::toTitleCase(k), "kernel"))
    rug(heights$Height_m)}))
```

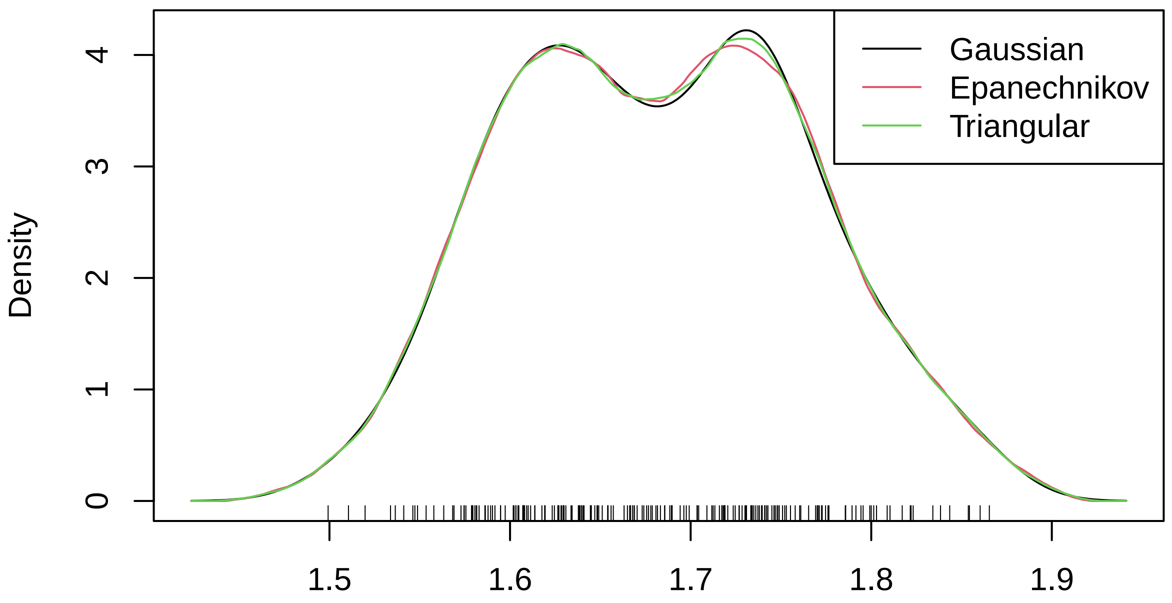






Alternatively on the same plot

```
kernel.estimates <- lapply(kernels, function(k) density(heights$Height_m, kernel = k))
x.seq <- kernel.estimates[[1]]$x
fhats <- lapply(kernel.estimates, "[", "y")
xlims <- range(x.seq)
ylims <- c(-1e-2, max(unlist(fhats) + 1e-2))
plot(c(), ty = "n", xlim = xlims, ylim = ylims,
     xlab = "", ylab = "Density")
for (k in seq_along(kernel.estimates))
    lines(kernel.estimates[[k]], col = k)
legend("topright", legend = tools::toTitleCase(kernels), col = 1:3, lty = "solid")
rug(heights$Height_m)
```



2.3 Probability calculation

Compute an estimate of the probability that a person is taller than 1.7m using a kernel density estimator.

```r
# Use the approximate and integrate functions
density.est <- density(heights[["Height_m"]])
f <- approxfun(density.est)
prob <- integrate(f, 1.7, max(density.est$x))
prob
```

```
## 0.4314333 with absolute error < 3.1e-05
```

2.4 MLE

If we assume the heights data follows a gaussian distribution, what are the maximum likelihood estimates of the parameters (i.e. the mean μ and standard deviation σ)? *Hint*: See `?mle` from the `stats4` library.

```r
library(stats4)
negativelogLikelihood <- function(m, s) {
    if (s < 0)
        return(Inf)
    -sum(log(dnorm(heights$Height_m, mean = m, sd = s)))
}
mle.fit <- mle(negativelogLikelihood, list(m = 1, s = 1))
mle.fit
```

```
##
## Call:
## mle(minuslogl = negativelogLikelihood, start = list(m = 1, s = 1))
##
## Coefficients:
##          m          s
## 1.68177704 0.08083764
```

3 Bandwidth selection

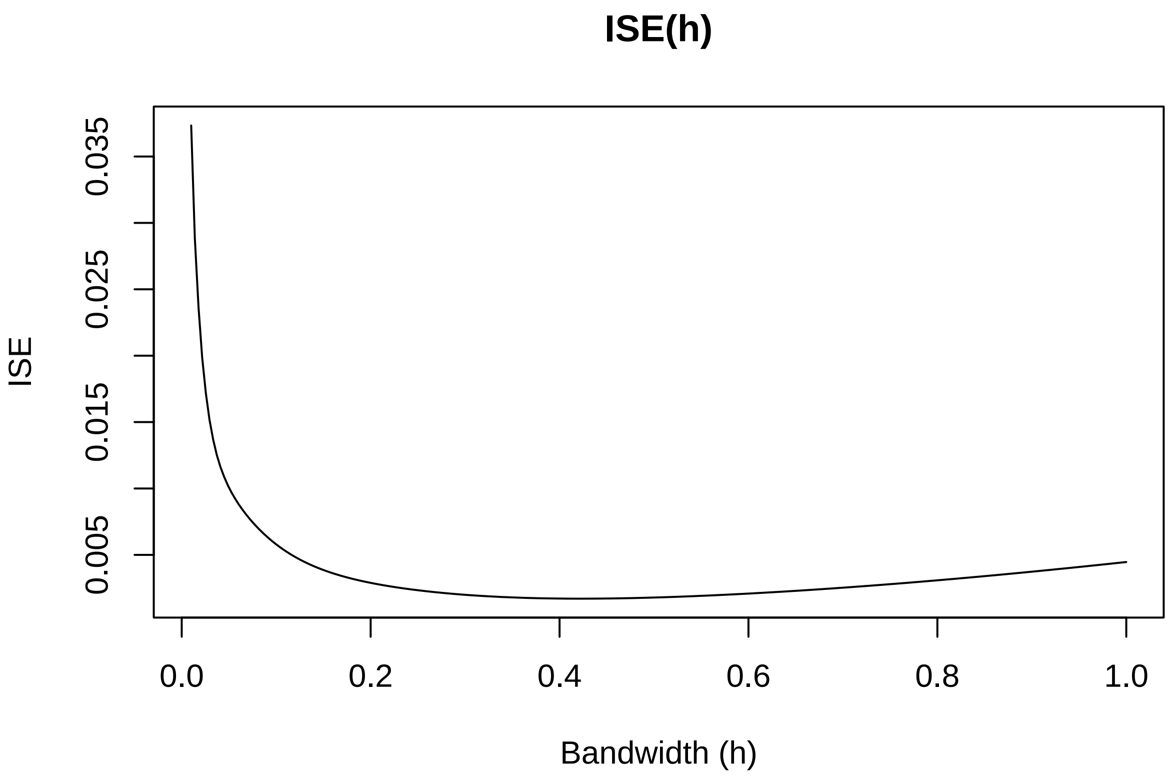
3.1 Integrated square error

Simulate 100 observations from a standard normal distribution. Using this sample conduct your investigation to determine the best bandwidth using the `density` function and computing the ISE (Integrated Square Error). The integrated square error is given by

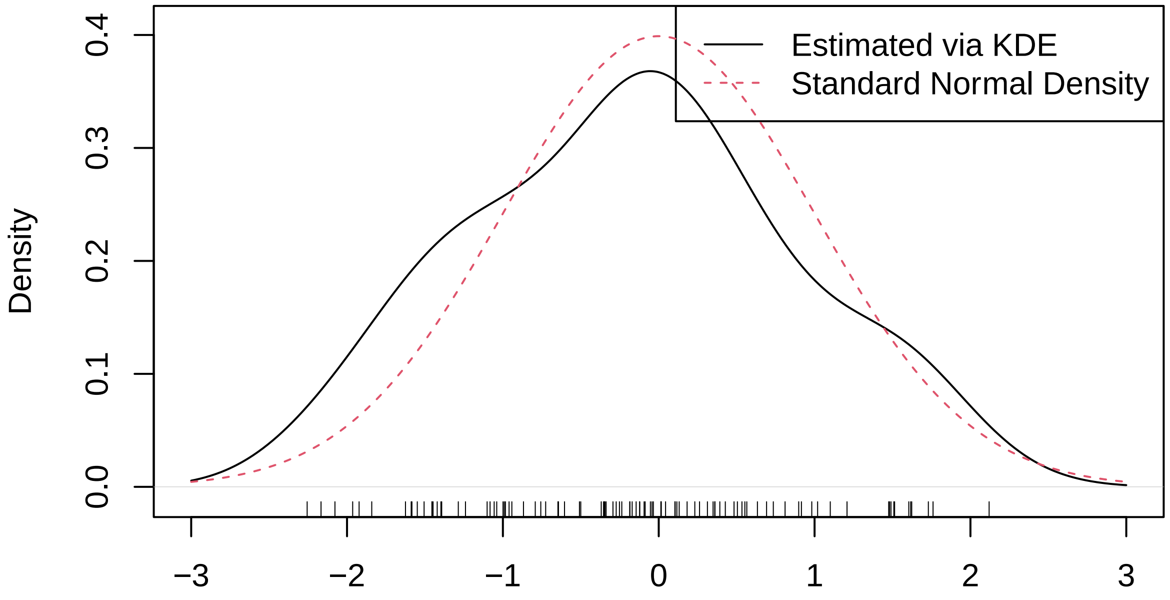
$$ISE(h) = \int_{-\infty}^{\infty} (\widehat{f}(x) - f(x))^2 \, dx$$

To do this, create grid of potential bandwidth values and compute the density estimator and validate it against the known standard normal density measuring its distance by the ISE. I would recommend using the arguments `n = 512, from = -3, to = 3` in `density` for appropriate validation. Create two plots. One of your bandwidth grid against the ISE and plot your final optimal estimate against the true standard normal density for comparison.

```r
set.seed(2021)
z <- rnorm(100)
h.grid <- seq(from = 0.01, to = 1, length.out = 256)
density.estimates <- lapply(h.grid, function(h) density(z, bw = h, n = 512, from = -3, to = 3))
ise <- function(f, fhat) mean((fhat - f)^2)
ise.h <- sapply(density.estimates, function(d) ise(dnorm(d$x), d$y))
plot(h.grid, ise.h, type = "l", main = "ISE(h)",
     xlab = "Bandwidth (h)", ylab = "ISE")
```



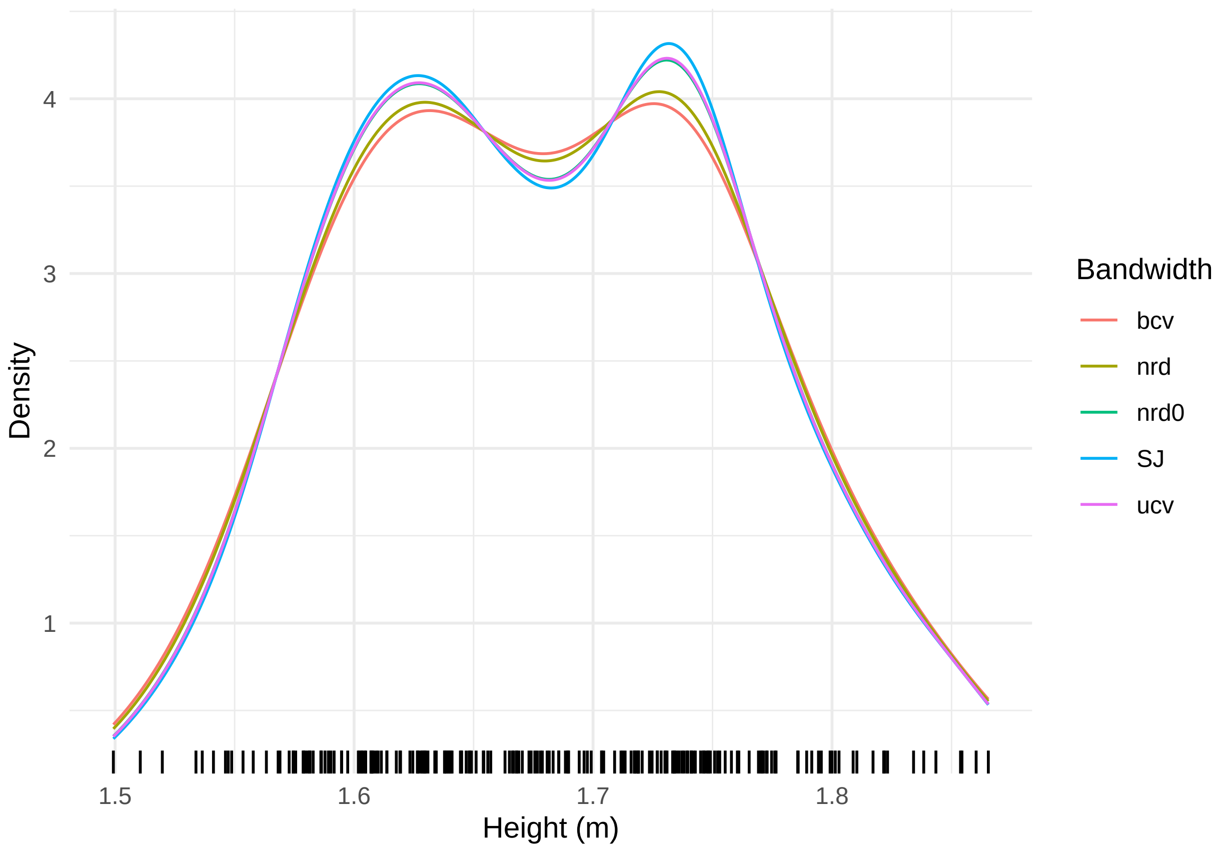
```r
opt.ind <- which.min(ise.h)
opt.h <- h.grid[opt.ind]
plot(density.estimates[[opt.ind]], main = "",
     xlab = "", ylab = "Density", ylim = c(-1e-2, dnorm(0) + 1e-2))
rug(z)
legend("topright", legend = c("Estimated via KDE",
                                "Standard Normal Density"),
       col = 1:2, lty = 1:2)
x <- density.estimates[[1]]$x
lines(x, dnorm(x), lty = 2, col = 2)
```



3.2 Bandwidth selection algorithms

For the heights dataset, explore the bandwidth selection algorithms given in the stats package (see for example the common documentation of some of the `? stats::bw.ucv`).

```
bw.methods <- c("nrd0", "nrd", "ucv", "bcv", "SJ")
rang <- range(heights$Height_m)
densities <- vapply(bw.methods, function(x) suppressWarnings(density(heights$Height_m, bw = x, from = ra
dat <- densities |> as.data.frame() |>
  gather(key = "Bandwidth", value = "density") |>
    mutate(x = rep(seq(from = rang[1], to = rang[2], length.out = 512), length(bw.methods)))
ggplot(dat) +
    geom_line(data = dat, aes(x = x, y = density, color = Bandwidth)) +
    theme_minimal() +
    geom_rug(data = heights, aes(x = Height_m)) +
    labs(x = "Height (m)", y = "Density")
```



The selected bandwidths are all quite similar in this dataset with the Sheather Jones bandwidth favouring a smaller bandwidth and the biased cross validation selected bandwidth favouring the largest bandwidth. In all cases there appears to be bimodal structure in the data. Perhaps indicative of two groups of different heights (perhaps male and female cohorts).

3.3 Bandwidth selection app

Using the heights data or some simulated data, create a Shiny app that allows the user to view the histogram of the data and computed density estimator.

See and compile Rmd file on Canvas for example Shiny app