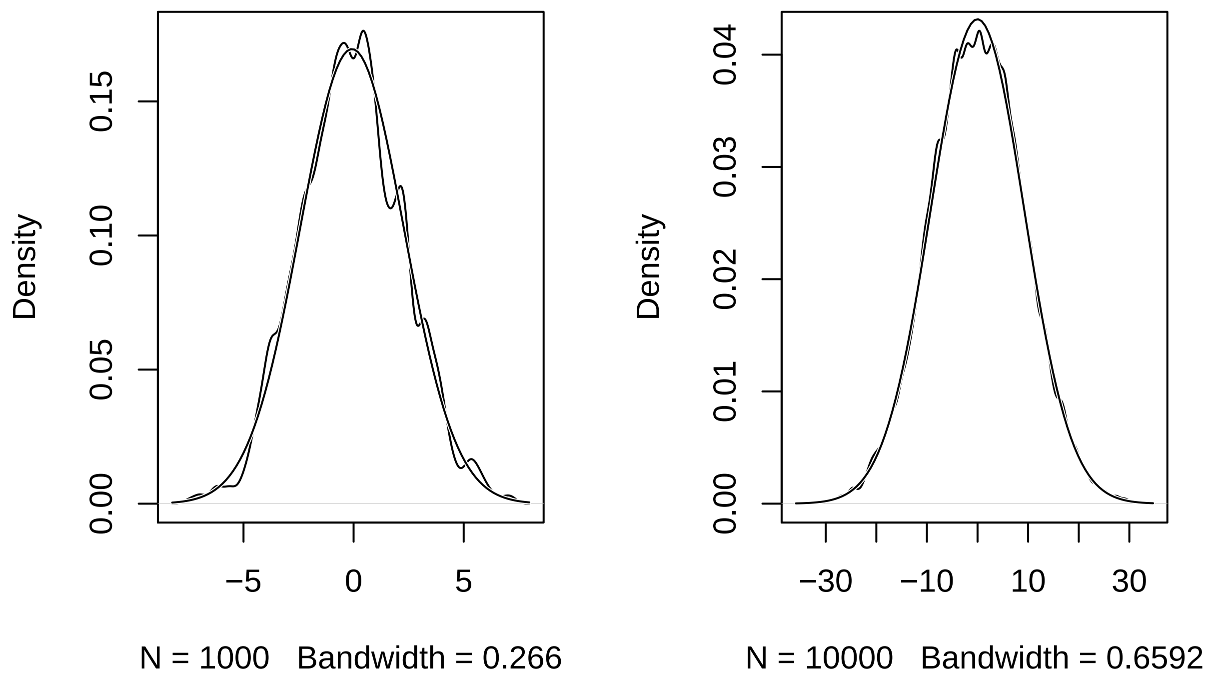
4 - Linear Models

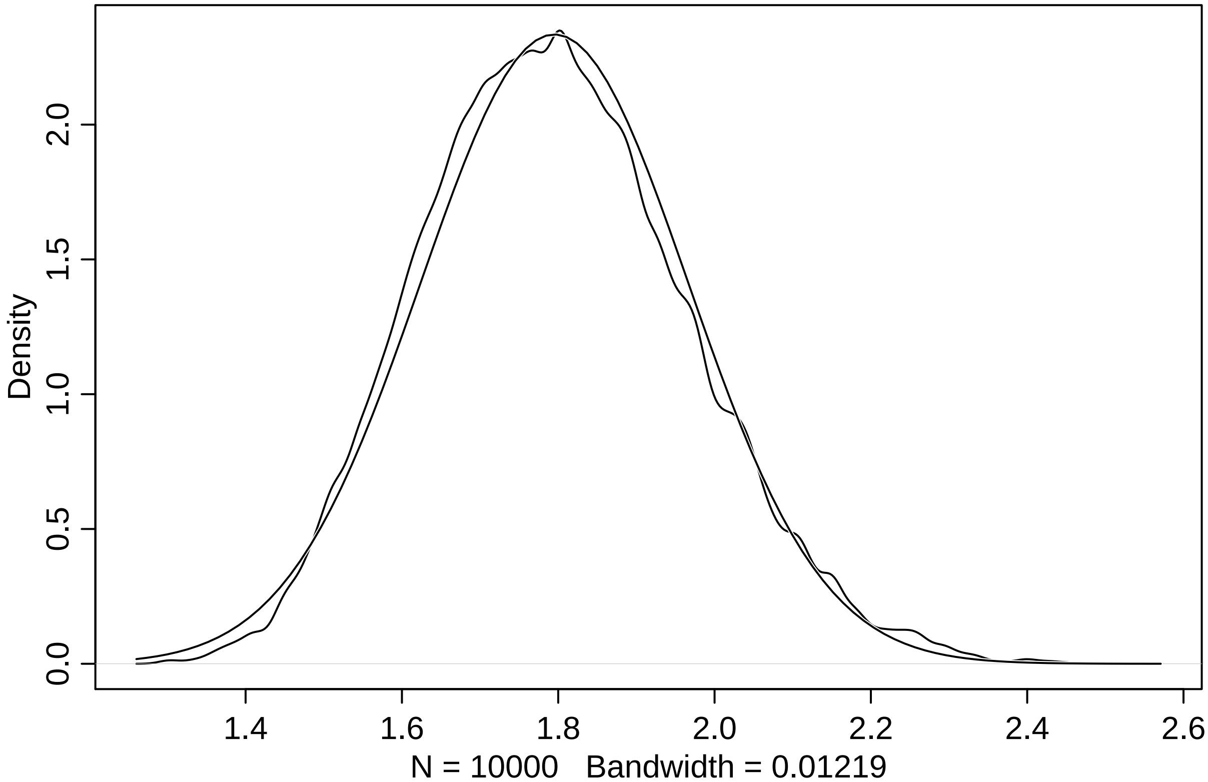
4.1.1. Normal by addition

```r
# 4.1
pos <- replicate(1000, sum(runif(16, -1, 1)))
par(mfrow=c(1, 2))
dens(pos, norm.comp = T)
dens(replicate(10000, sum(runif(256, -1, 1))), norm.comp = T)
```

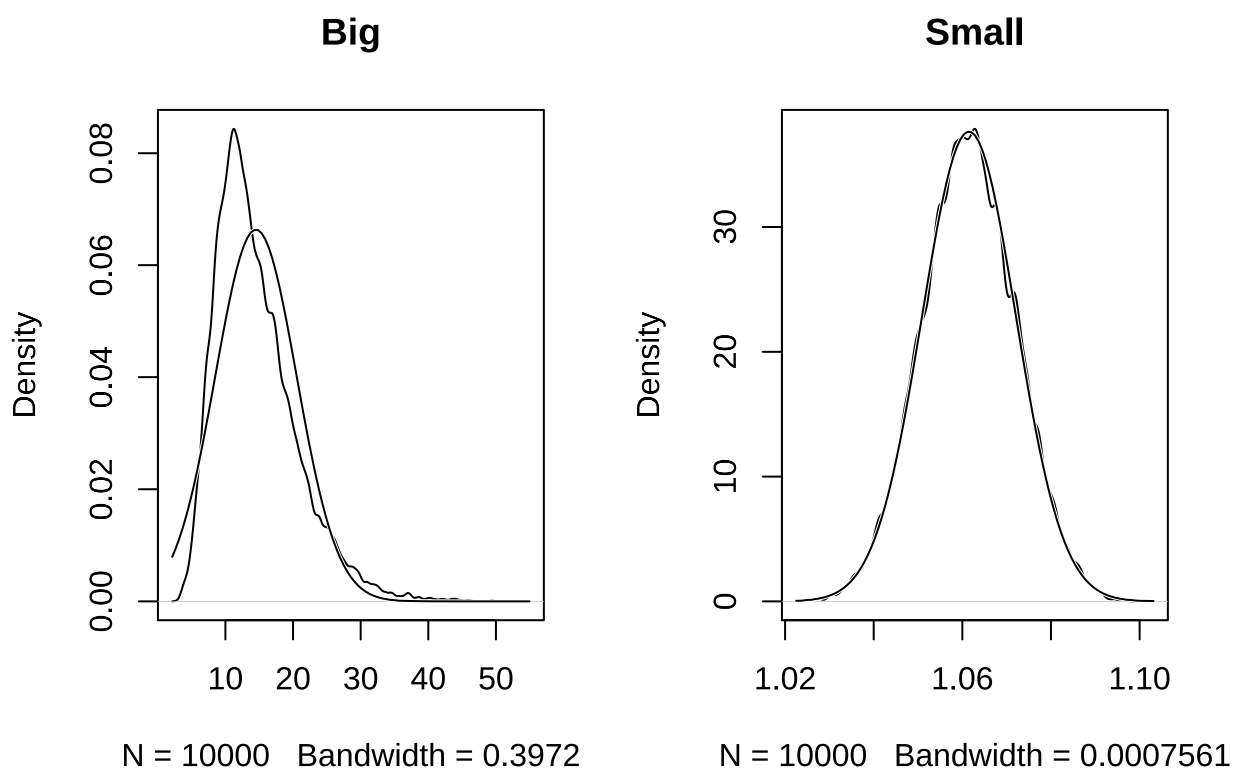


4.1.2. Normal by multiplication

```r
# 4.2
dens(replicate(1e4, prod(1 + runif(12, 0, 0.1))), norm.comp = T)
```

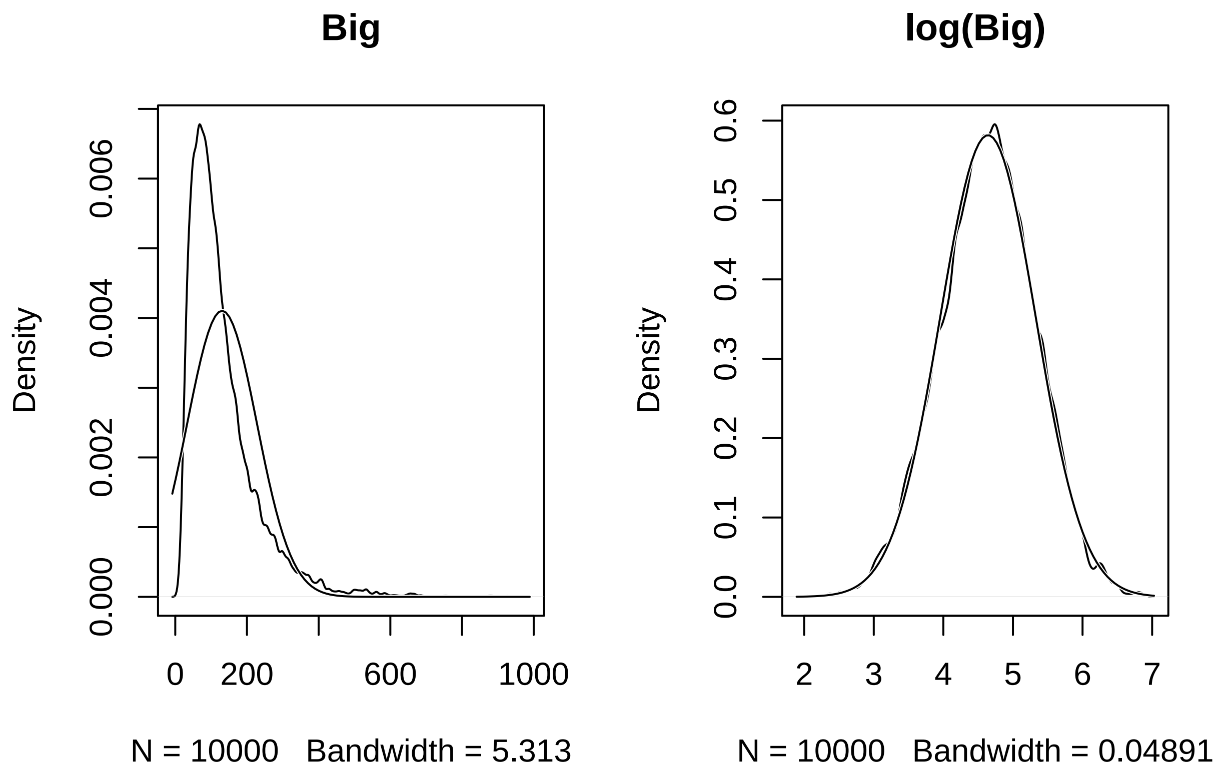



```r
# 4.4
big <- replicate(1e4, prod(1 + runif(12, 0, 0.5)))
small <- replicate(1e4, prod(1 + runif(12, 0, 0.01)))
par(mfrow=c(1, 2))
dens(big, norm.comp = T, main = "Big")
dens(small, norm.comp = T, main = "Small")
```



Normal by log-multiplication

```
# 4.5
big <- replicate(1e4, prod(1 + runif(12, 0, 1)))
log_big <- log(big)
par(mfrow=c(1, 2))
dens(big, norm.comp = T, main = "Big")
dens(log_big, norm.comp = T, main = "log(Big)")
```



4.3 A Gaussian model of height

```r
# 4.7
library(rethinking)
data(Howell1)
d <- Howell1
```

```r
# 4.8
str(d)
```

```
## 'data.frame':    544 obs. of  4 variables:
##  $ height: num  152 140 137 157 145 ...
##  $ weight: num  47.8 36.5 31.9 53 41.3 ...
##  $ age   : num  63 63 65 41 51 35 32 27 19 54 ...
##  $ male  : int  1 0 0 1 0 1 0 1 0 1 ...
```

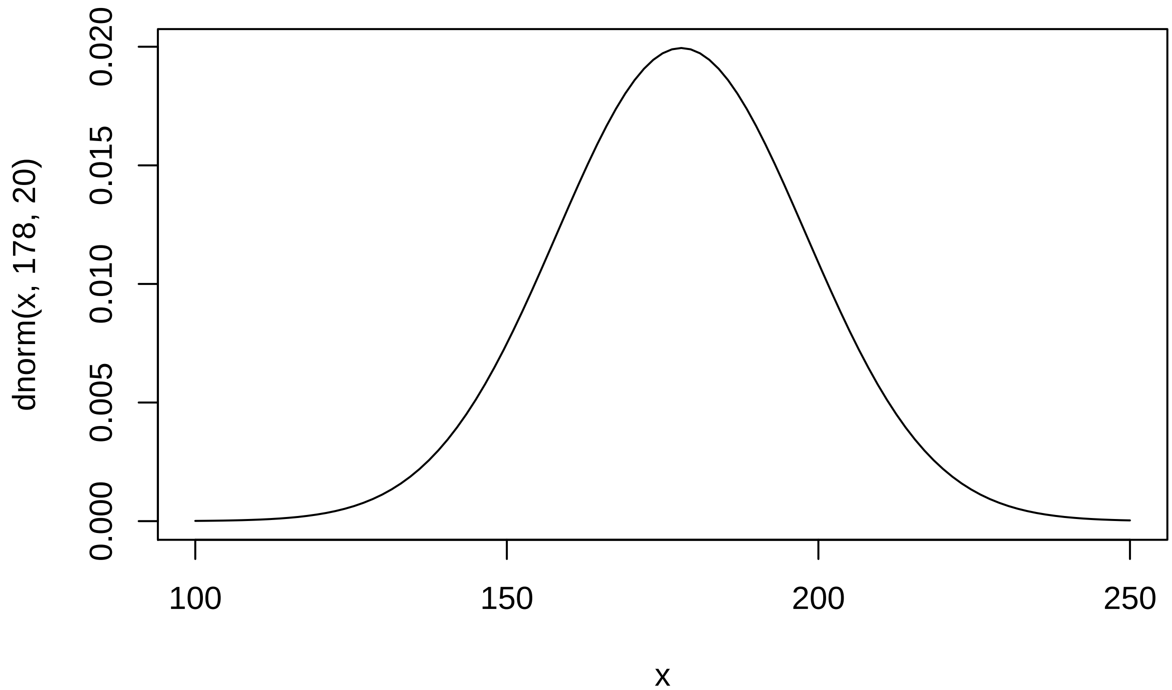
We want heights of adults only (352 rows):

```r
# 4.10
d2 <- d[d$age >= 18, ]
```

4.3.2 The model

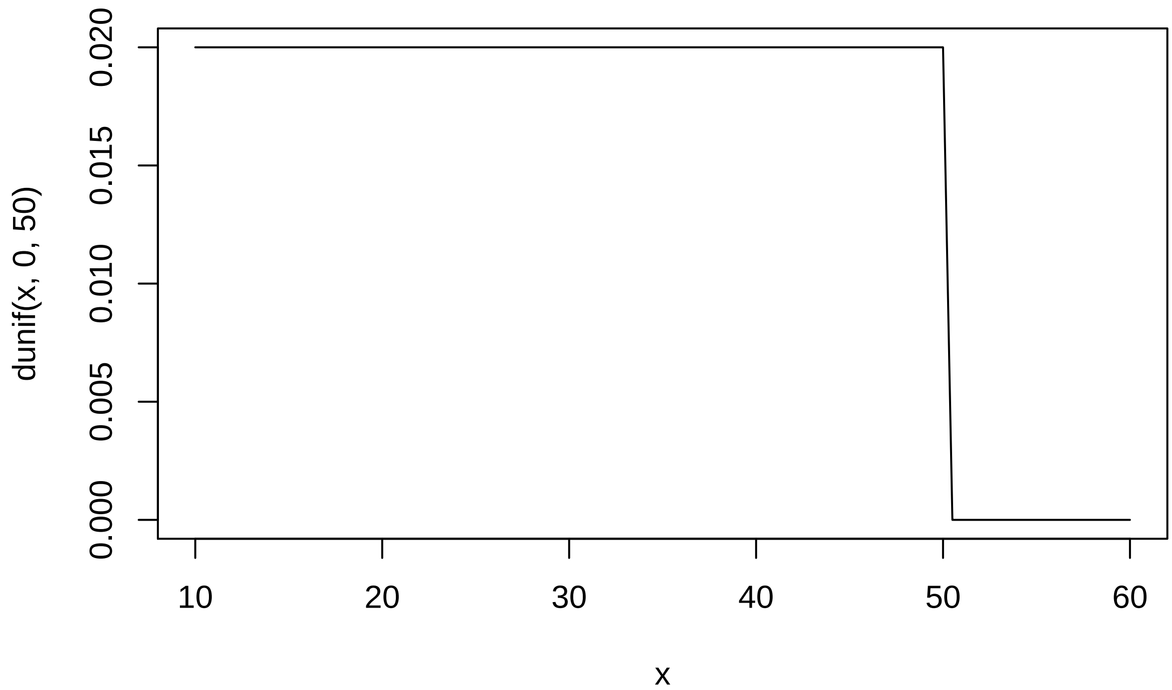
Height mean:

```r
# 4.11
curve(dnorm(x, 178, 20), from=100, to=250)
```

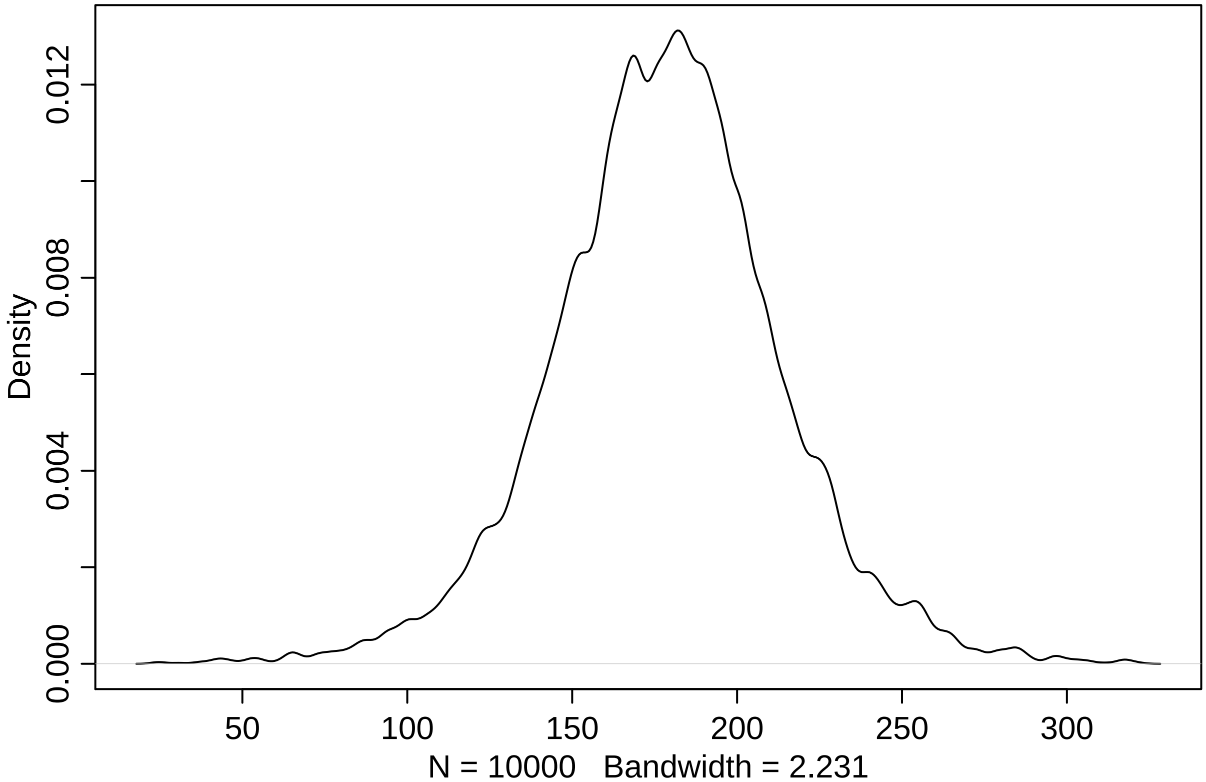


Height standard deviation:

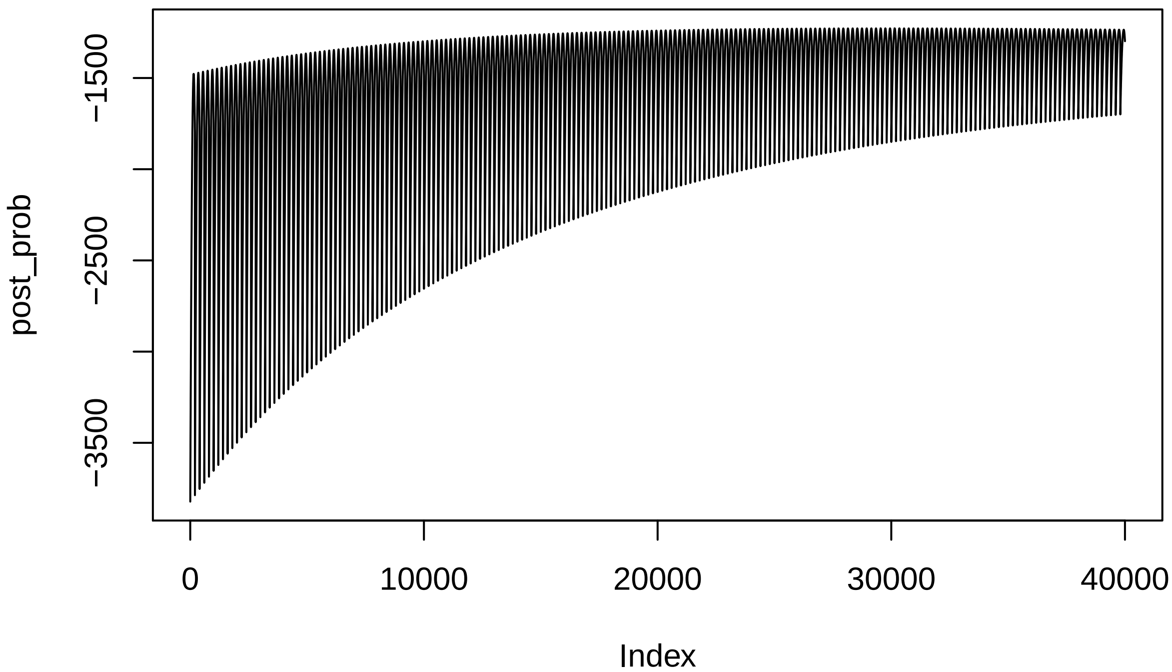
```r
# 4.12
curve(dunif(x, 0, 50), from=10, to=60)
```




```r
# 4.13
sample_mu <- rnorm(1e4, 178, 20)
sample_sigma <- runif(1e4, 0, 50)
prior_h <- rnorm(1e4, sample_mu, sample_sigma)
dens(prior_h)
```

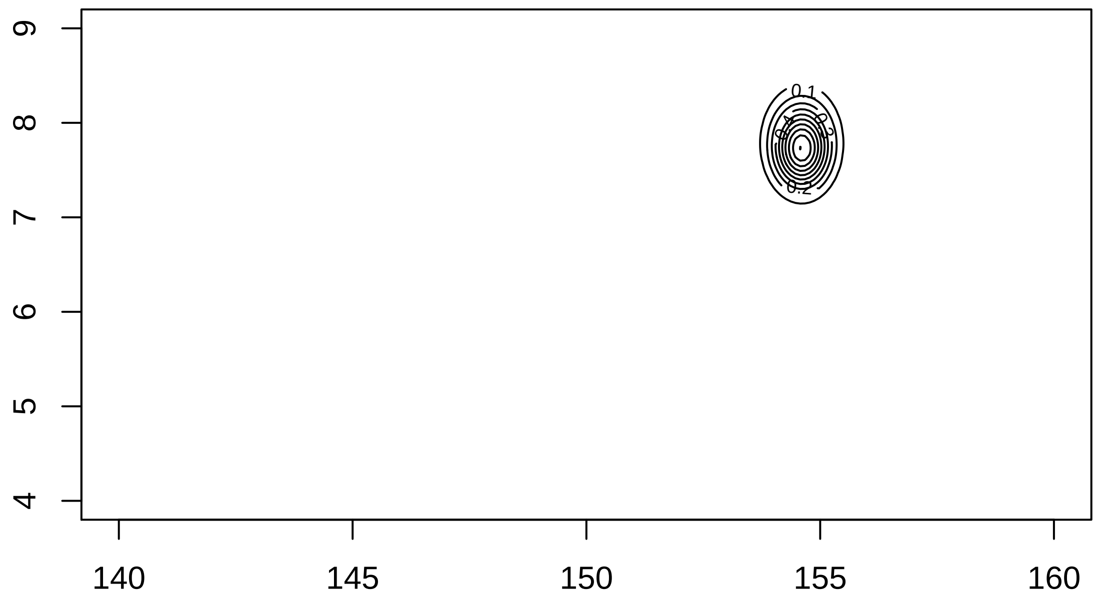



```r
# 4.14
mu_list <- seq(from=140, to=160, length.out=200)
sigma_list <- seq(from=4, to=9, length.out=200)
post <- expand.grid(mu=mu_list, sigma=sigma_list)
post_ll <- sapply(1:nrow(post), function(i) sum(dnorm(
  d2$height,
  mean=post$mu[i],
  sd=post$sigma[i],
  log=T
)))
post_prob <- post_ll + dnorm(post$mu, 178, 20, T) + dunif(post$sigma, 0, 50, T)
plot(post_prob, type="l")
```

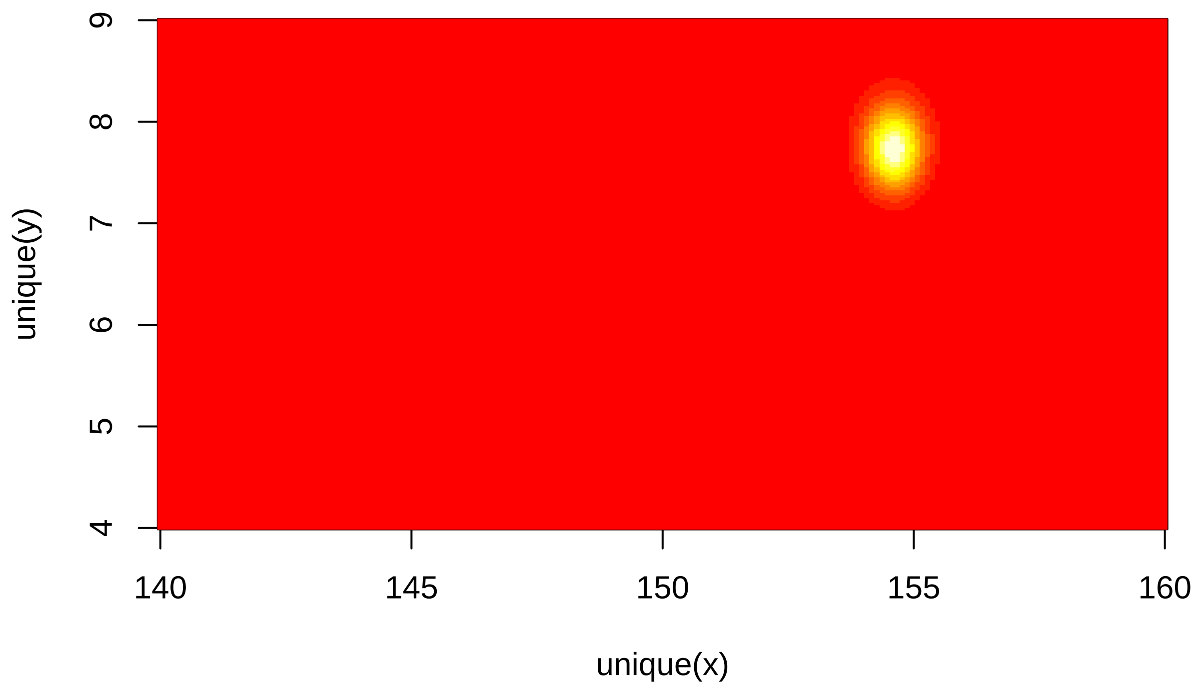



```r
post_prob <- exp(post_prob - max(post_prob))
```

```r
# 4.15
contour_xyz(post$mu, post$sigma, post_prob)
```



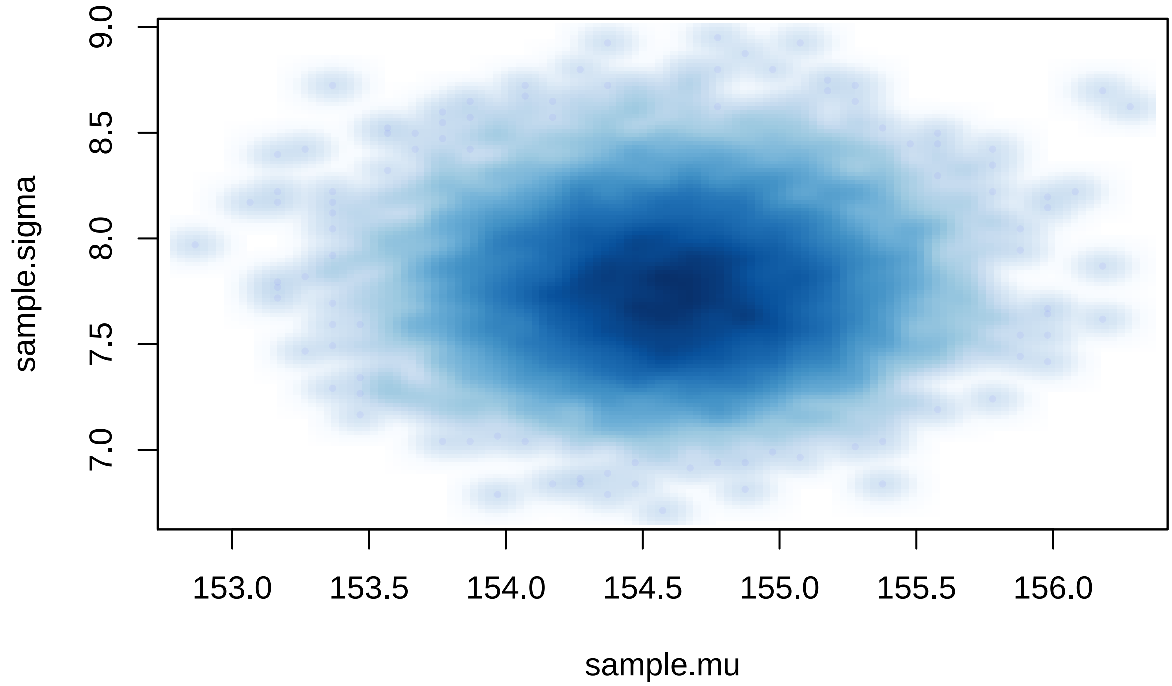

```
# 4.16
image_xyz(post$mu, post$sigma, post_prob)
```



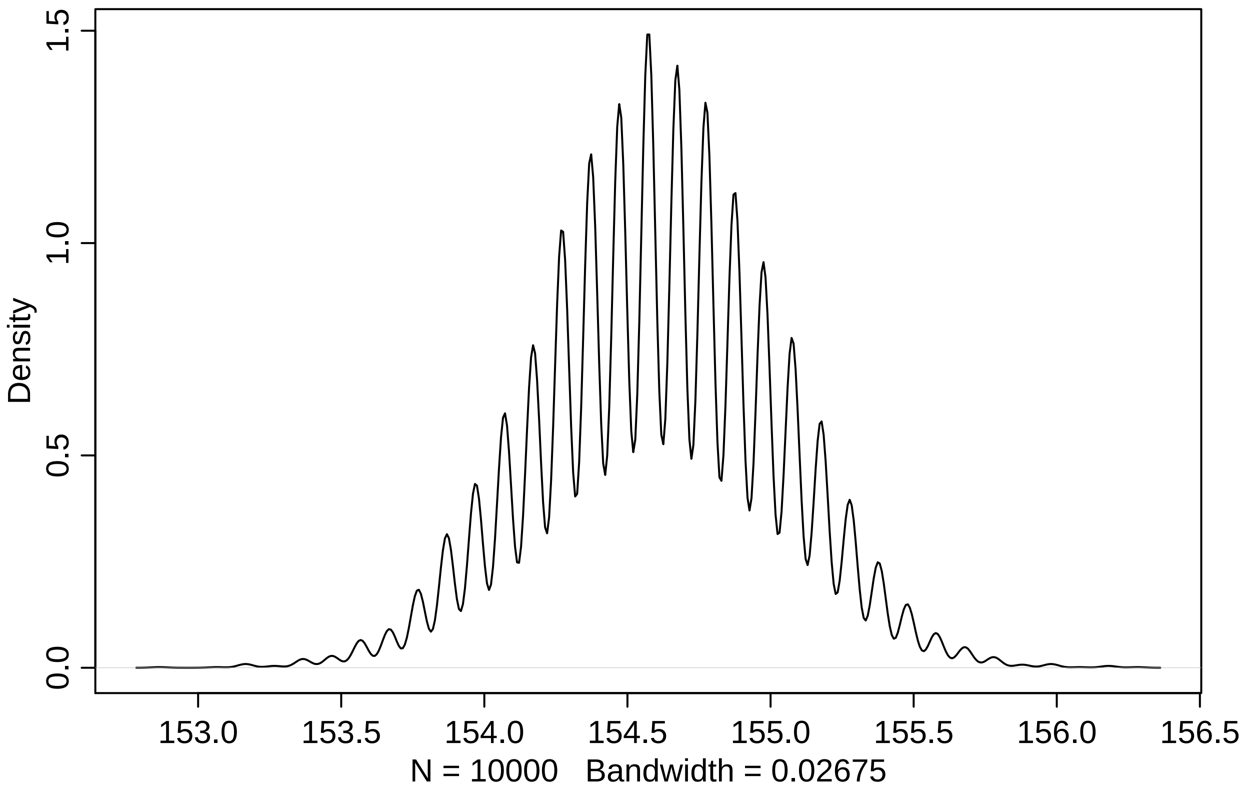
4.3.4 Sampling from the posterior

```
# 4.17
sample.rows <- sample(1:nrow(post), size=1e4, replace = T, prob = post_prob)
sample.mu <- post$mu[sample.rows]
sample.sigma <- post$sigma[sample.rows]
```

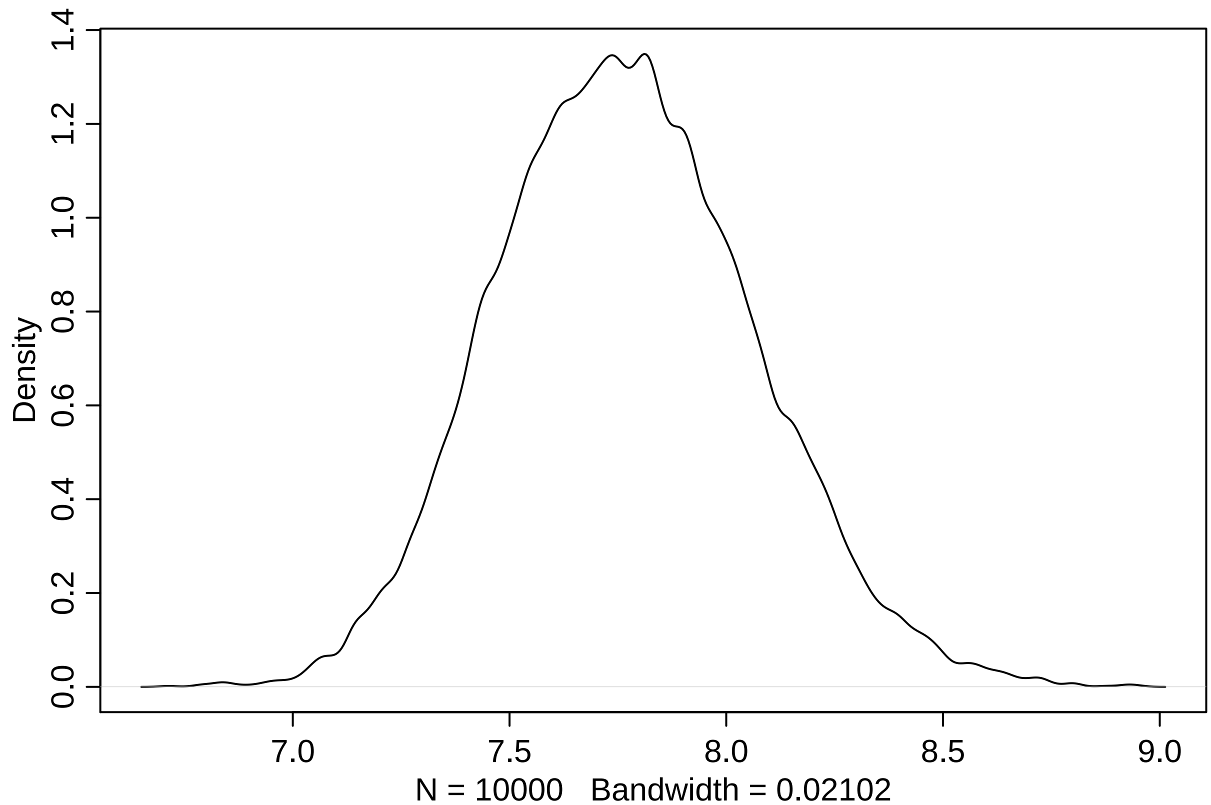
```
# 4.18
smoothScatter(sample.mu, sample.sigma, cex=0.5, pch=16, col=col.alpha(rangi2, 0.1))
```




```
# 4.19
dens(sample.mu)
```




```r
dens(sample.sigma)
```




```r
# 4.20
HPDI(sample.mu)
```

```
##     |0.89    0.89|
## 153.8693 155.1759
```

```r
HPDI(sample.sigma)
```

```
##     |0.89    0.89|
## 7.266332 8.195980
```

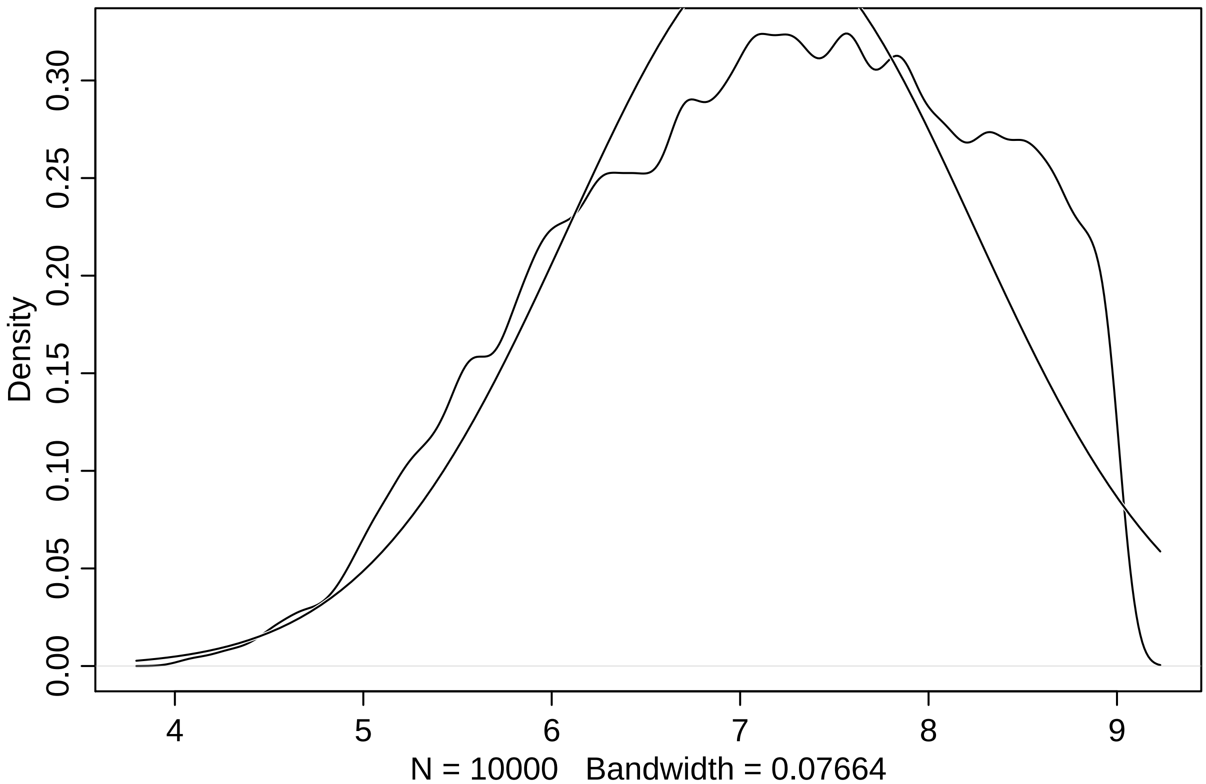
Smaller Sample

To illustrate the posterior is not always Guassian in shape.

```r
# 4.22
d3 <- sample(d2$height, size=10)

small.post_ll <- sapply(1:nrow(post), function(i) sum(dnorm(d3, mean=post$mu[i], sd=post$sigma[i], log='

small.post_product <- small.post_ll + dnorm(post$mu, 178, 20, T) + dunif(post$sigma, 0, 50, T)

small.post_proba <- exp(small.post_product - max(small.post_product))

small.sample.rows <- sample(1:nrow(post), size=1e4, replace = T, prob=small.post_proba)

small.sample.mu <- post$mu[small.sample.rows]
```

```r
small.sample.sigma <- post$sigma[small.sample.rows]
```

```r
# 4.23
dens(small.sample.sigma, norm.comp = T)
```



4.3.5. Fitting the model with *map*

map finds the values of μ and σ that maximize the posterior probability.