

# STAT 206 Homework 7

Xin Feng(Vanessa)

11/15/2019

Due Monday, November 18, 5:00 PM

**General instructions for homework:** Homework must be submitted as pdf file, and be sure to include your name in the file. Give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Each answer must be supported by written statements as well as any code used. (Examining your various objects in the “Environment” section of RStudio is insufficient – you must use scripted commands.)

## Part I - Beverton-Holt model

The dataset at [<http://www.faculty.ucr.edu/~jfflegal/fish.txt>] contains 40 annual counts of the numbers of spawners  $S$  and recruits  $R$  in a salmon population. The units are thousands of fish. Spawners are fish that are laying eggs. Spawners die after laying eggs. Recruits are fish that enter the catchable population.

The classic **Beverton-Holt** model for the relationship between spawners and recruits is

$$R = \frac{1}{\beta_1 + \beta_2/S}, \quad \beta_1 > 0, \beta_2 > 0$$

where  $R$  and  $S$  are the number of recruits and spawners respectively.

Consider the problem of maintaining a sustainable fishery. The total population abundance will only stabilize if  $R = S$ . The total population will decline if fewer recruits are produced than the number of spawners who died producing them. If too many recruits are produced, the population will also decline eventually because there is not enough food for them all. Thus, only a balanced level of recruits can be sustained indefinitely in a stable population. This stable population level is the point where the  $45^\circ$  line intersects the curve relating  $R$  and  $S$ . In other words, it is the  $N$  such that

$$N = \frac{1}{\beta_1 + \beta_2/N}.$$

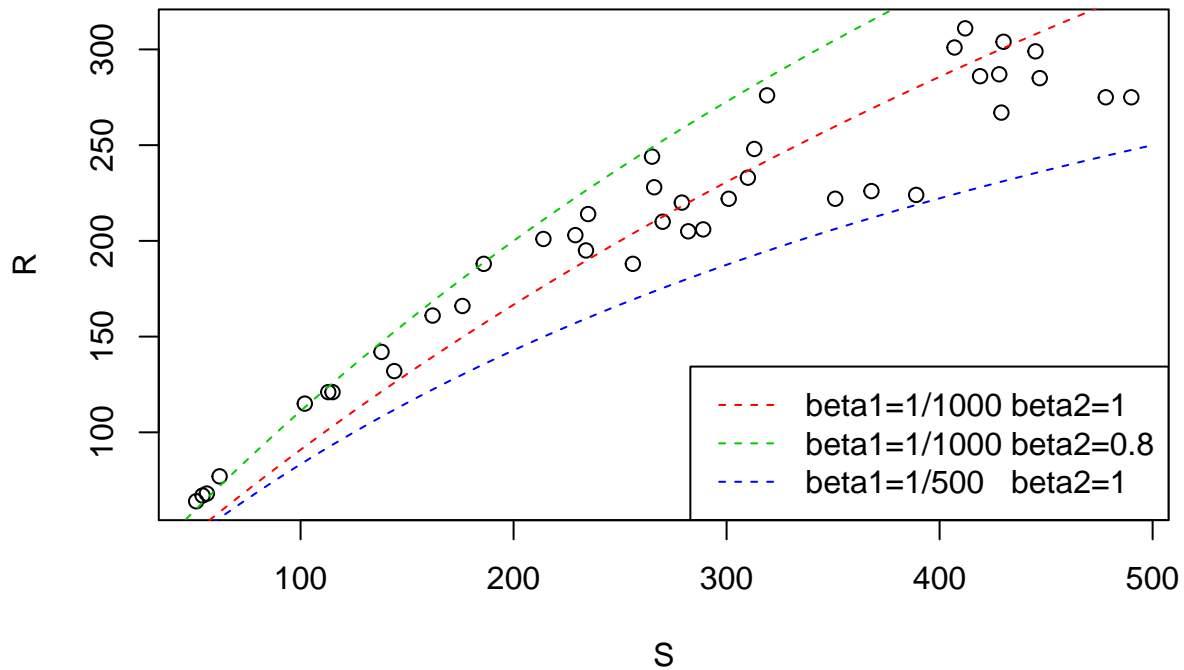
Solving for  $N$  we see that the stable population level is  $N = (1 - \beta_2)/\beta_1$ .

```
data <- read.table('http://www.faculty.ucr.edu/~jfflegal/fish.txt', stringsAsFactors = FALSE)
# View(data)
data1 <- data.matrix(data[-1,])
```

1. Make a scatterplot of the data and overlay the Beverton-Holt curve for a couple different choices of  $\beta_1$  and  $\beta_2$ .

```
Beho <- function(beta1=1, beta2=1, S){
  R <- 1/(beta1+beta2/S)
  return(R)
}
plot(data1[,2], data1[,1], xlab='S', ylab='R', main='scatterplot of S and R')
curve(Beho(1/1000,1,S=x), 0, 500, col=2, lty=2, add=TRUE)
curve(Beho(1/1000,0.8,S=x), 0, 500, col=3, lty=2, add=TRUE)
curve(Beho(1/500,1,S=x), 0, 500, col=4, lty=2, add=TRUE)
legend('bottomright', legend=c('beta1=1/1000 beta2=1', 'beta1=1/1000 beta2=0.8',
                              'beta1=1/500 beta2=1'), lty=c(2,2,2), col=c(2,3,4))
```

## scatterplot of S and R



2. The Beverton-Holt model can be found by transforming  $R \mapsto (1/R)$  and  $S \mapsto (1/S)$ . That is,

$$(1/R) = \beta_1 + \beta_2(1/S).$$

This is a linear model with response variable  $(1/R)$  and covariate  $(1/S)$ . Use least squares regression to fit this model to the fish dataset.

```
data2 <- 1/data1
frac_S <- data2[,2]
frac_R <- data2[,1]
plot(frac_S, frac_R, xlab='1/S', ylab='1/R', main='Scatterplot of 1/S and 1/R')
cor(frac_S, frac_R)
```

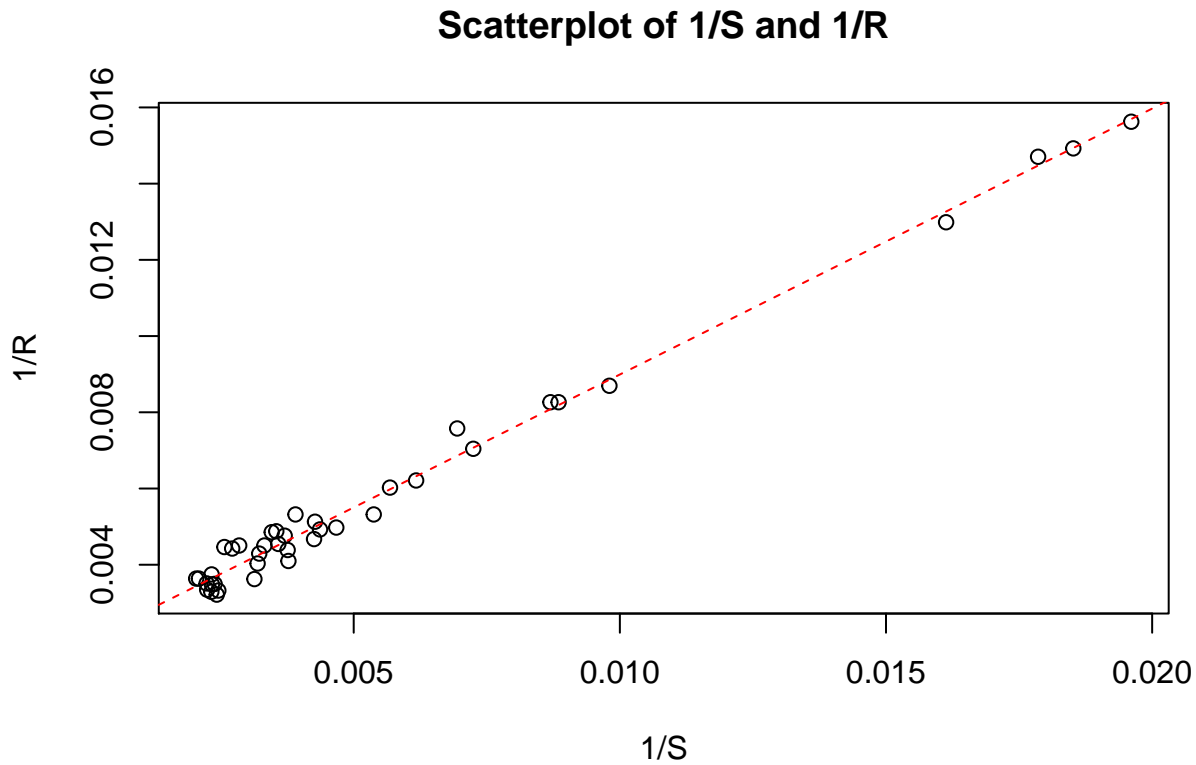
```
## [1] 0.9948862
```

```
fit <- lm(frac_R~frac_S)
summary(fit)
```

```
##
## Call:
## lm(formula = frac_R ~ frac_S)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.776e-04 -2.403e-04 -1.903e-05  1.755e-04  7.166e-04
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 2.013e-03 8.216e-05 24.50 <2e-16 ***
## frac_S      6.978e-01 1.149e-02 60.72 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0003362 on 38 degrees of freedom
## Multiple R-squared:  0.9898, Adjusted R-squared:  0.9895
## F-statistic: 3687 on 1 and 38 DF, p-value: < 2.2e-16
```

```
abline(fit, lty=2, col=2)
```



- Find an estimate for the stable population level, where  $R = S$  in the Beverton-Holt model.

```
# When R=S, 1/R=1/S, 1/R=beta1+beta2*(1/R)
# 1/R=beta1/(1-beta2)
beta1 <- fit$coefficients[[1]]
beta2 <- fit$coefficients[[2]]
Get.N <- function(beta1, beta2) (1-beta2)/beta1
stable_value <- Get.N(beta1, beta2)
cat('The estimate of N =', stable_value, '\n')
```

```
## The estimate of N = 150.0976
```

- Use the bootstrap to obtain the sampling distribution and standard error for the stable population level. Use the bootstrap to construct a 95% confidence interval for the stable population level.

```

M <- 1e+3
n <- nrow(data2)
boot_N <- vector(mode='numeric', length=0)
for(i in 1:M){
  index <- sample(1:n, n, replace=TRUE)
  fit <- lm(data2[index,1]~data2[index,2])
  beta1 <- fit$coefficients[[1]]
  beta2 <- fit$coefficients[[2]]
  boot_N[i] <- Get.N(beta1, beta2)
}
sd_N <- sd(boot_N)
cat("The standard error for the stable population level is ", sd_N, '\n')

```

```
## The standard error for the stable population level is 3.719844
```

```

# 95% confidence interval
t.test(boot_N)

```

```

##
## One Sample t-test
##
## data: boot_N
## t = 1275.7, df = 999, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 149.8367 150.2983
## sample estimates:
## mean of x
## 150.0675

```

```

# Method II
mcse <- function(y) sd(y)/sqrt(length(y))
interval <- mean(boot_N) + c(-1,1)*qt(1-0.05/2, length(boot_N-1))*mcse(boot_N)
cat("95% confidence interval of N is ", interval, '\n')

```

```
## 95% confidence interval of N is 149.8367 150.2983
```

## Part II - Snowfall accumulations

The data set `buffalo` at [<http://www.faculty.ucr.edu/~jflegal/buffalo.txt>] contains annual snowfall accumulations in Buffalo, NY from 1910 to 1973.

```

buffalo_snow <- read.table('http://www.faculty.ucr.edu/~jflegal/buffalo.txt')
is.numeric(buffalo_snow[1,])

```

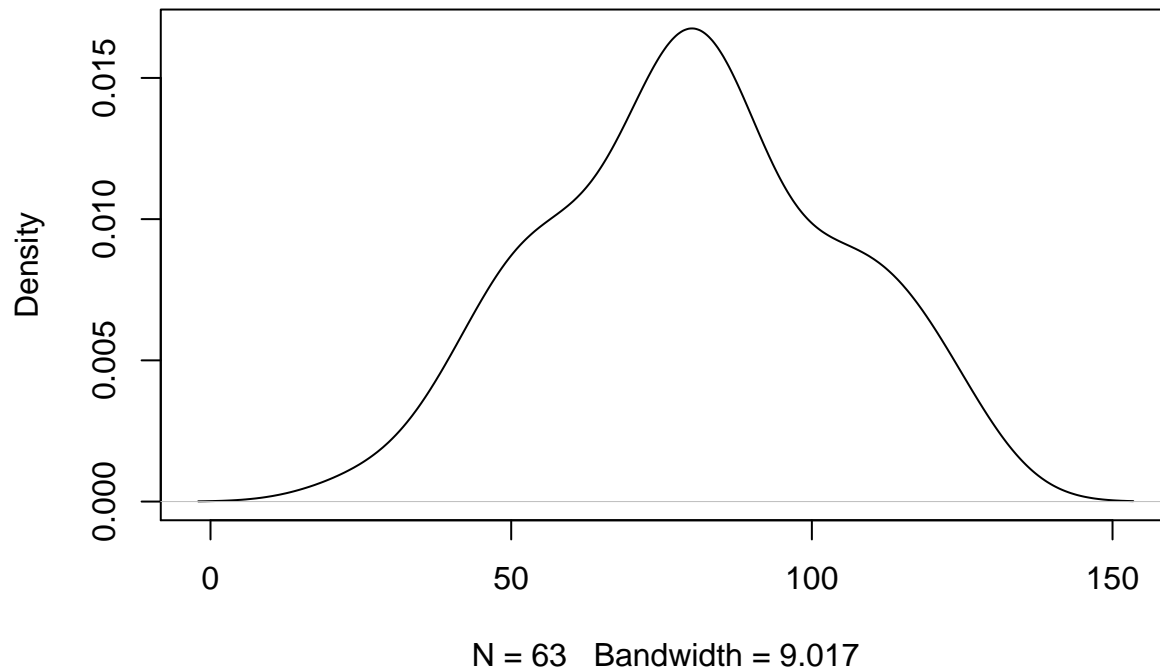
```
## [1] TRUE
```

```
# View(buffalo)
```

5. Construct kernel density estimates of the data using the Gaussian and Epanechnikov kernels.

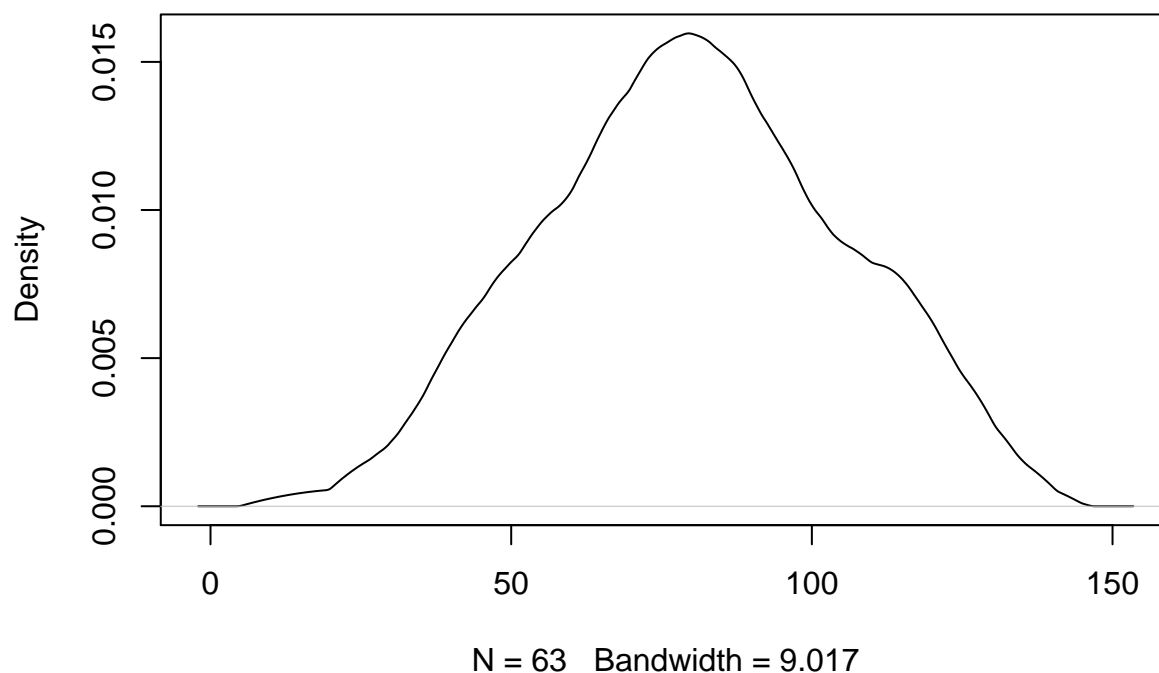
```
d_snow1 <- density(buffalo_snow[,1], bw='SJ', kernel="gaussian")  
plot(d_snow1, main="Kernel density estimates with Gaussian kernel")
```

### Kernel density estimates with Gaussian kernel



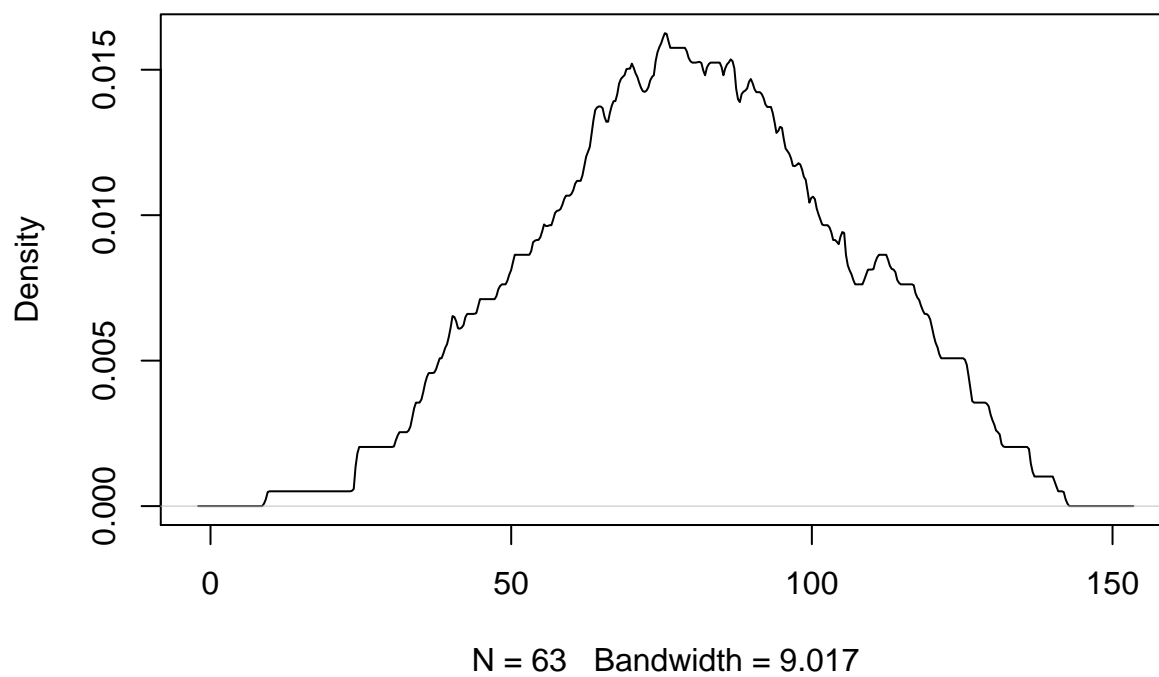
```
d_snow2 <- density(buffalo_snow[,1], bw='SJ', kernel="epanechnikov")  
plot(d_snow2, main="Kernel density estimates with Epanechnikov kernel")
```

## Kernel density estimates with Epanechnikov kernel



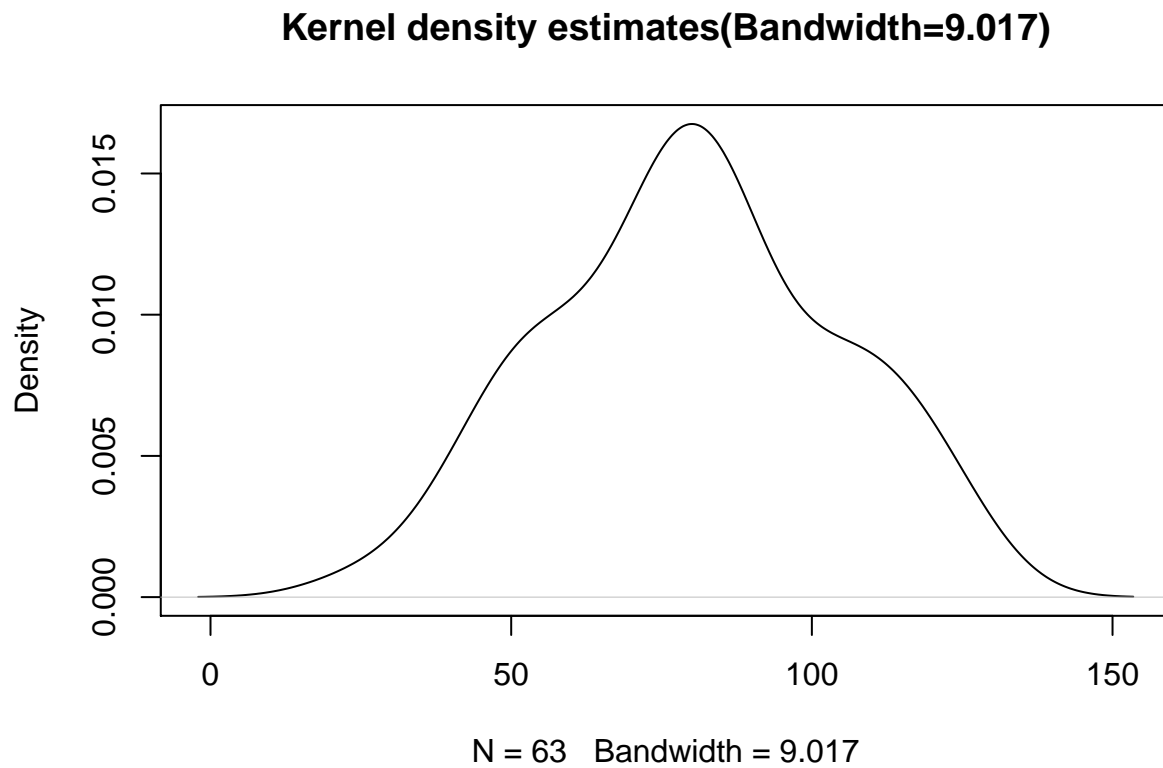
```
d_snow3 <- density(buffalo_snow[,1], bw='SJ', kernel="rectangular")  
plot(d_snow3, main="Kernel density estimates with Rectangular kernel")
```

## Kernel density estimates with Rectangular kernel



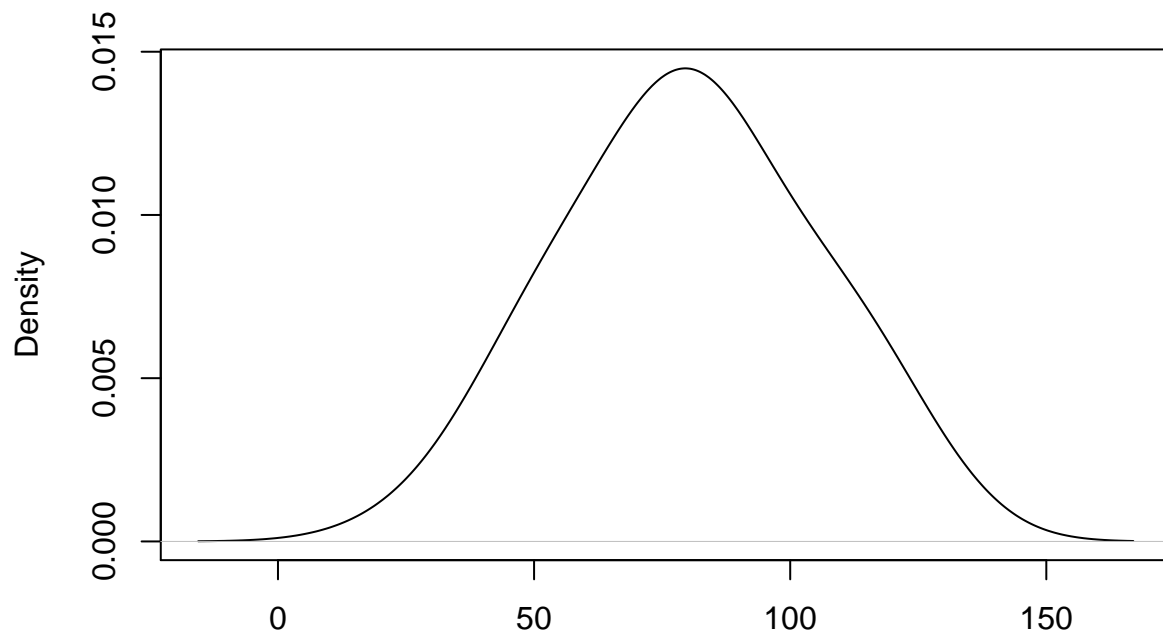
6. Compare the estimates for different choices of bandwidth.

```
d_snow4 <- density(buffalo_snow[,1], bw="SJ", adjust=1)
plot(d_snow4, main="Kernel density estimates(Bandwidth=9.017)")
```



```
d_snow5 <- density(buffalo_snow[,1], bw="SJ", adjust=1.5)
plot(d_snow5, main="Kernel density estimates(Bandwidth=13.53)")
```

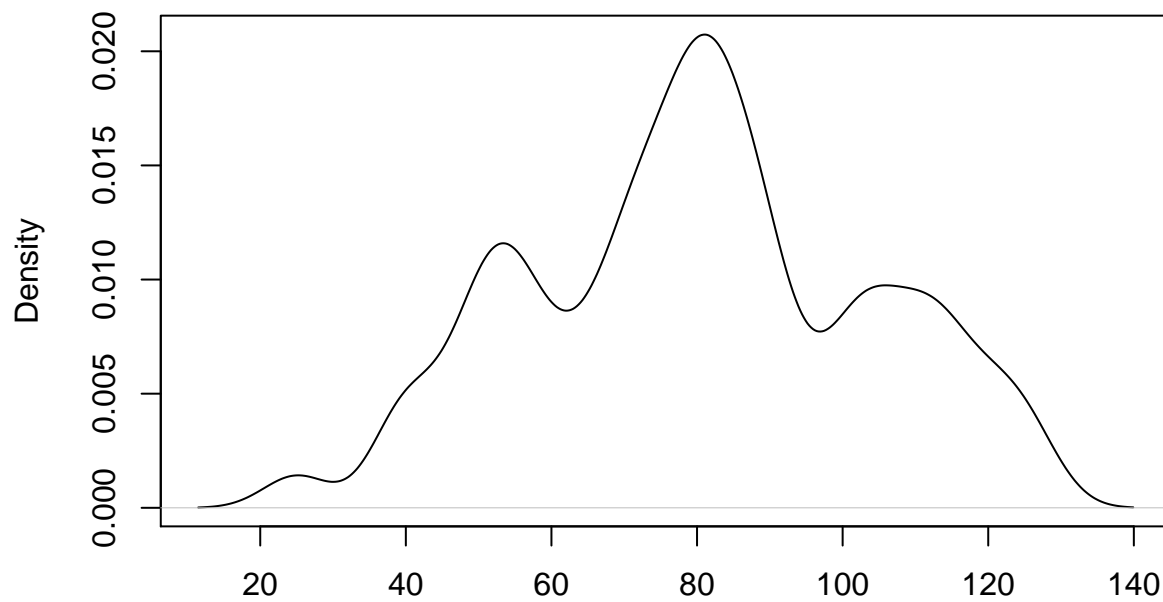
### Kernel density estimates(Bandwidth=13.53)



N = 63 Bandwidth = 13.53

```
d_snow6 <- density(buffalo_snow[,1], bw="SJ", adjust=0.5)
plot(d_snow6, main="Kernel density estimates(Bandwidth=4.509)")
```

### Kernel density estimates(Bandwidth=4.509)



N = 63 Bandwidth = 4.509



*# When the bandwidth greater, Kernel density curve smoother.*

7. Is the estimate more influenced by the type of kernel or the bandwidth?

*# According to the Question 5 and Question 6, the bandwidth is more influenced the  
# estimate.*

*# For KDE curves of multiple data points, the shape of the curve formed is not  
# closely related to the selected kernel function due to the waveform synthesis  
# between adjacent peaks.*