STA 410S/2102S: Test 1, February 22, 2005, 1:10 - 2:00 pm
The questions are of equal value; you are permitted one aid sheet (8.5 x 11).

1. *Simple linear regression*: Consider the simple linear model

$$y_i = \beta_0 + \beta_1(x_i - \bar{x}) + \epsilon_i, \quad i = 1, \ldots, n \quad \epsilon_i \sim (0, \sigma^2) \tag{1}$$

where $x_1, \ldots x_n$ are fixed constants and we assume ϵ_i are independently distributed. This is a special case of the linear regression model, and the least squares estimates of β_0 and β_1 are

$$\hat{\beta}_0 = \bar{y}, \qquad \hat{\beta}_1 = \frac{\Sigma(y_i - \bar{y})(x_i - \bar{x})}{\Sigma(x_i - \bar{x})^2}. \tag{2}$$

Show that $\hat{\beta}_1$ has the following properties under model (1):

$$E(\hat{\beta}_1) = \beta_1, \qquad \mathrm{var}(\hat{\beta}_1) = \sigma^2/\Sigma(x_i - \bar{x})^2. \tag{3}$$

2. *Simple linear regression cont'd*: I wrote a program to simulate the performance of $\hat{\beta}_1$ when model (1) is not true. The simulated data comes from a model with unequal variances:

$$y_i = \beta_0 + \beta_1(x_i - \bar{x}) + \epsilon_i, \quad i = 1, \ldots, n \quad \epsilon_i \sim (0, \sigma_i^2) \tag{4}$$

where $x_1, \ldots x_n$ are fixed constants and we continue to assume ϵ_i are independently distributed. The program, `sim.lse`, calls `lse`, a program that assumes model (1) is true and computes $\hat{\beta}_1$ and an estimate of $\mathrm{var}(\widehat{\beta_1})$ using formula (3).

These programs are reproduced below without comments: everywhere you see `#n` supply a comment that helps to explain how the program works.

```
lse <- function(x,y){
n <- length(x)
if(length(y) != n){stop("x and y must have same length")} #1
if(n <= 1){stop ("need at least two observations")}
xdev <- x - mean(x)
if(all(xdev)==0){stop("x's are all equal")}
b1 <- sum(xdev*(y-mean(y)))/sum(xdev*xdev) #2
b0 <- mean(y)
sig2hat <- sum((y-b0-b1*xdev)^2)/(n-2) #3
vb1 <- sig2hat/sum(xdev^2) #4
list(b1,vb1)}

sim.lse <- function(Nsim,vecsig,n=10){
if(length(vecsig)!= n){stop("need to supply n values for sigma^2_i")}
x<- 1:n    #5
beta0 <- 3
beta1 <- 2  #6
```

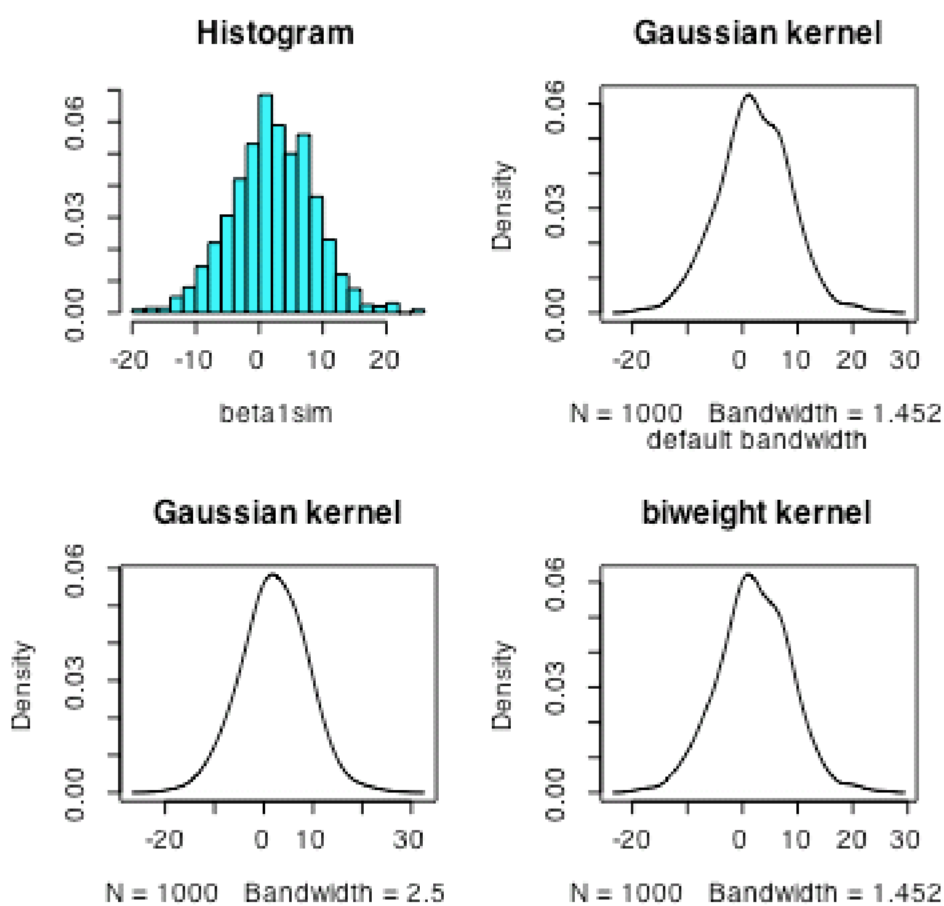
```
b1 <- rep (0, Nsim)
vb1 <- rep (0,Nsim)
for(k in 1:Nsim){
epsilon <- rnorm(n)*sqrt(vecsig)  #7
y <- beta0 + beta1*(x-mean(x)) + epsilon
lseout <- lse(x,y)
b1[k] <- lseout[[1]]
vb1[k] <- lseout[[2]]  #8
}
list(mean(b1),var(b1),mean(vb1))} #9
```

3. I used a variation on this program to simulate the entire density of $\hat{\beta}_1$ using a kernel density estimator

$$\hat{f}(x) = \frac{1}{Nb} \sum_{k=1}^{N} K\left(\frac{x - \texttt{beta1sim}_k}{b}\right).$$

   (a) What are the roles of the function $K(\cdot)$ and the parameter b?

   (b) Below are examples of 3 density estimates with specific choices of $K(\cdot)$ and b. Which estimate of the density of $\hat{\beta}$ do you prefer and why?



4. The `abbey` dataset contains 31 determinations of nickel content in a rock sample. The values are:

```
> abbey
 [1]   5.2   6.5   6.9   7.0   7.0   7.0   7.4   8.0   8.0   8.0   8.0
[12]   8.5   9.0   9.0  10.0  11.0  11.0  12.0  12.0  13.7  14.0  14.0
[23]  14.0  16.0  17.0  17.0  18.0  24.0  28.0  34.0 125.0
```

Following the book I computed several summary statistics in R, as follows:

```
> mean(abbey)
[1] 16.00645
> median(abbey)
[1] 11
> unlist(hubers(abbey))
       mu         s
11.731514  5.258487
> unlist(hubers(abbey,k=2))
       mu         s
12.351117  6.105222
> unlist(hubers(abbey,k=1))
       mu         s
11.365392  5.567345
> unlist(huber(abbey))
      mu        s
11.55136  4.44780
> mad(abbey)
[1] 4.4478
> IQR(abbey)
[1] 7
```

(a) Explain to a non-statistician why all these estimates of 'mu' are different. Which one would you recommend?

(b) What are `mad(abbey)` and `IQR(abbey)` estimating?