Praphull Kumar
UID: 204 271 732

STATS 202A
Fall 2014
Homework 2

Output:

Figure 1: A plot of your kernel density estimate of the earthquake magnitudes, fˆ (m), from part 1b), along with your simulation-based 95% confidence bands from part 1d).

In Below Figure,
- Blue Color - 97.5% percentile
- Green Color - 2.5% percentile
- Black Color - Kernel Density estimate from Question 1.b

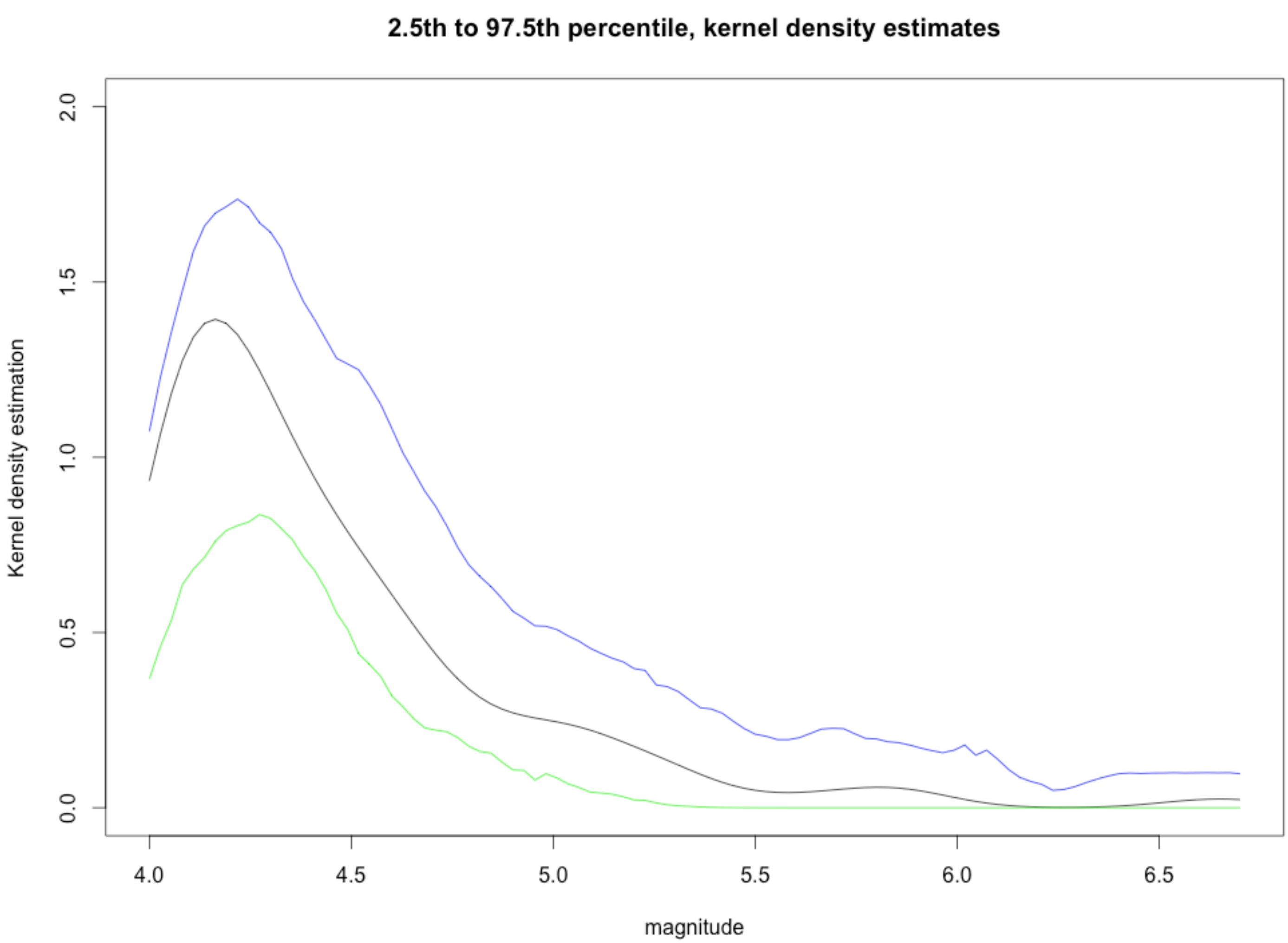


Figure 2: A plot of your 2-dimensional kernel smoothing of the earthquake locations, along with the points themselves and a legend, from part 1e).

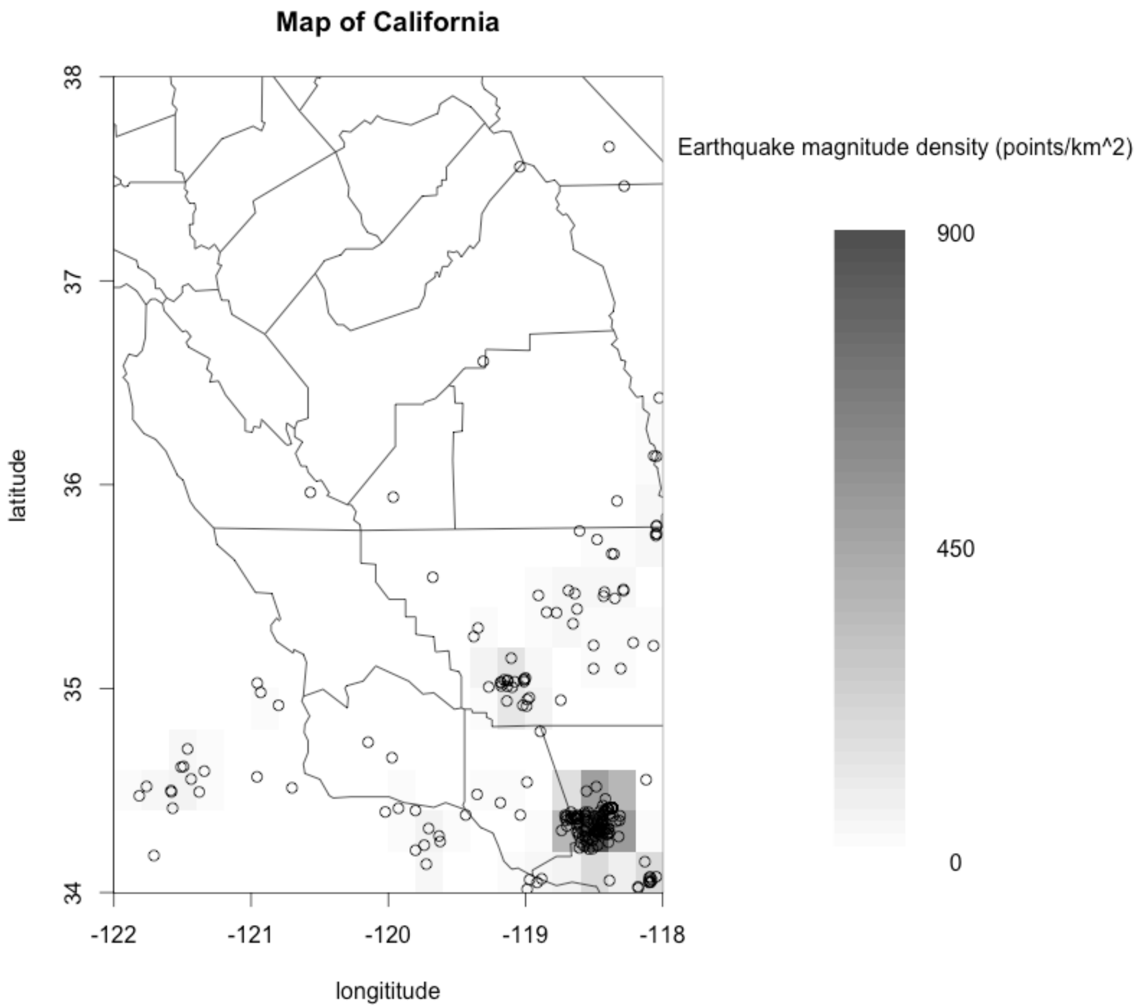
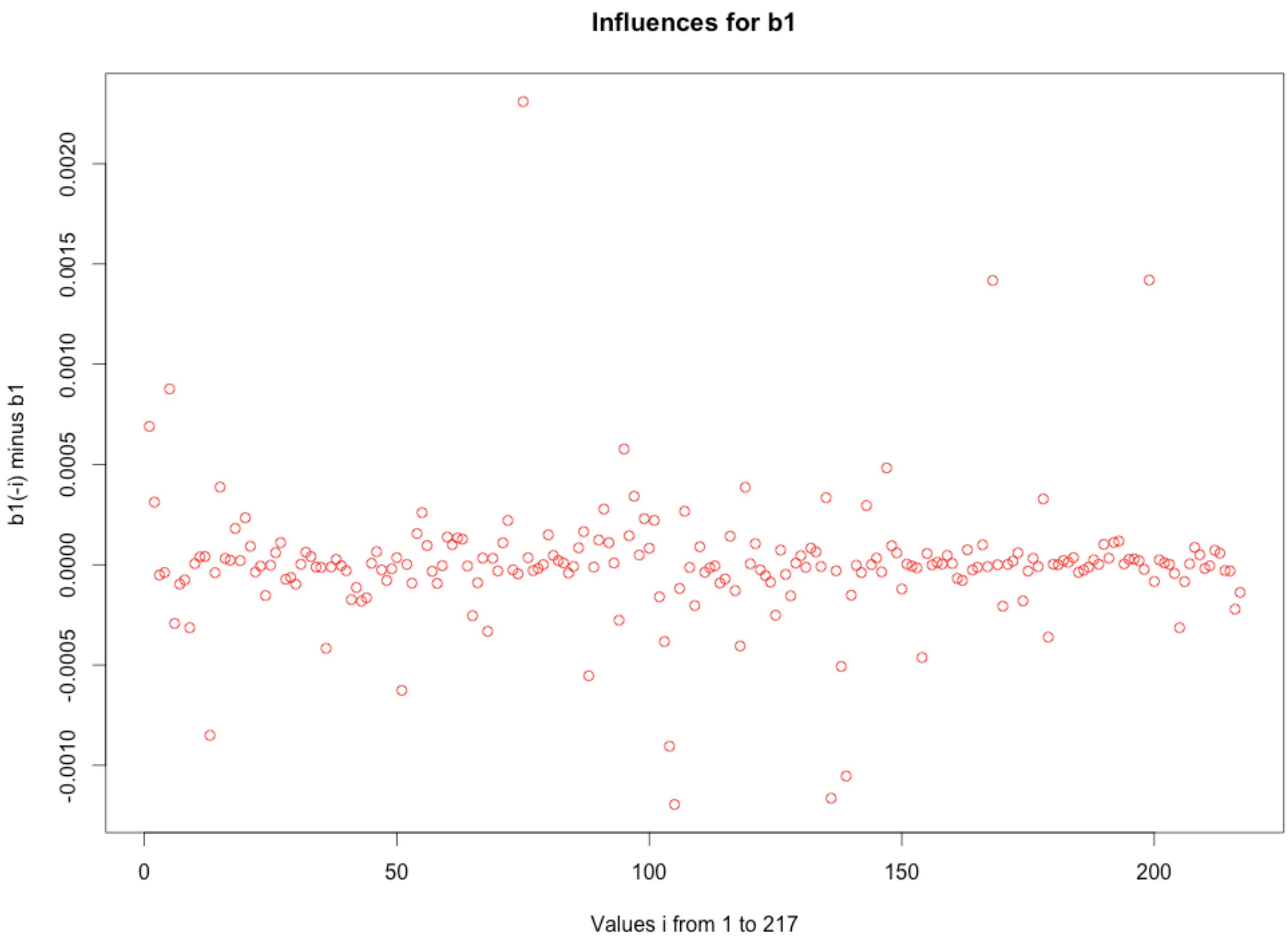


Figure 3: A plot of the influences, β̂ (-i) - β̂ , versus i, from part 2e).



Code:

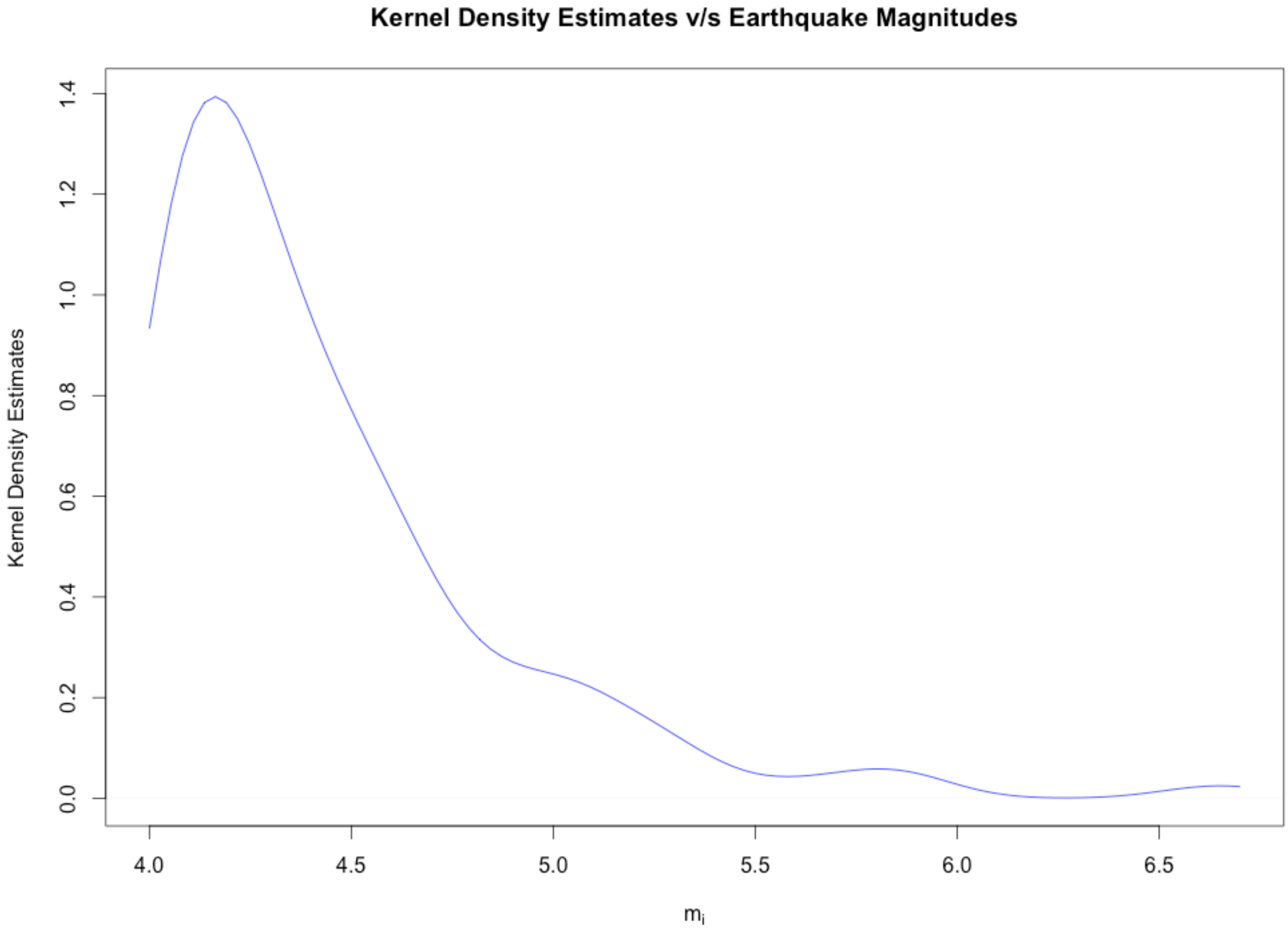
```
library(maps)
library(splancs)
library(mapproj)

#Question 1.a
#============
dataE = scan("SearchResults.txt", skip=3, nlines=232, what="char")
dataMatrixE = matrix(dataE, ncol=12, byrow=T)
colNamesE = scan("SearchResults.txt", skip=2, nlines=1, what="char")
colnames(dataMatrixE) = colNamesE

#Question 1.b
#============
mag_vec = as.numeric(as.vector(dataMatrixE[,'MAG']))
k_density_1b = density(mag_vec, bw="nrd", adjust=1, kernel="gaussian", n=100,
```

Praphull Kumar
UID: 204 271 732

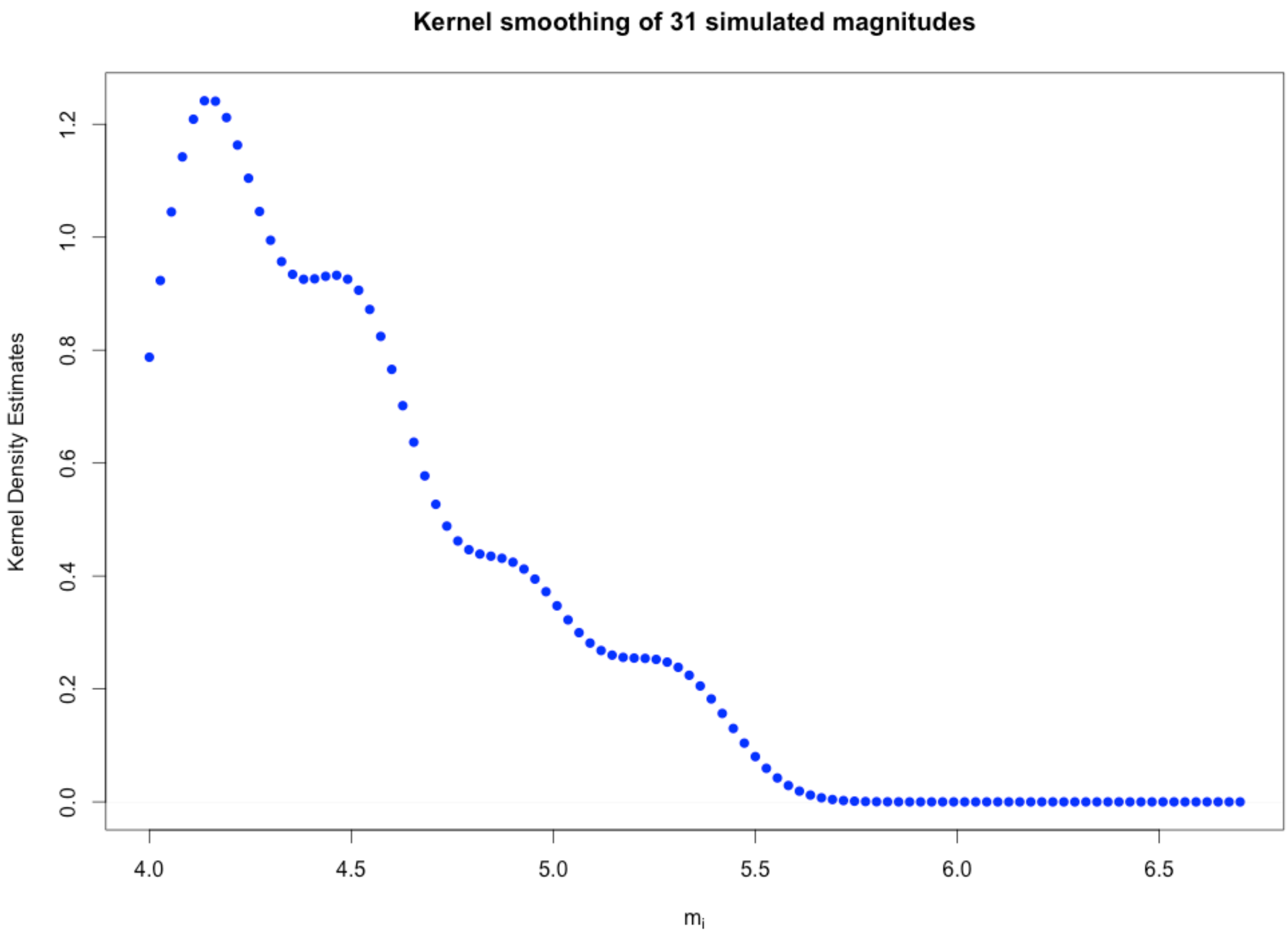
```
                       from=min(mag_vec), to=max(mag_vec))
plot(k_density_1b, type="l", xlab=expression("m"['i']),
     ylab="Kernel Density Estimates", pch=16, col="blue",
     main="Kernel Density Estimates v/s Earthquake Magnitudes")
```




```
#Question 1.c
#============
density_31_func = function () {
  c = max(k_density_1b[['y']])
  d = max(mag_vec) - min(mag_vec)
  b = c*d
  g = 1/d
  n = 31
  x = c()
  i=0
  while(i < n) {
    x0 = runif(1)*d + min(mag_vec)
    fx0 = density(mag_vec, bw="nrd", adjust=1, kernel="gaussian", n=1, from=x0, to=x0)
    if(runif(1) < fx0[['y']]/(b*g)) {
      i = i+1
```

```
      x[i] = x0
      if(i/100 == floor(i/100))
        cat(i, " ")
    }
  }
  kd = density(x, bw=k_density_1b[['bw']], adjust=1, kernel="gaussian", n = 100,
               from=min(mag_vec), to=max(mag_vec))
  return (kd)
}

kd_31_1c = density_31_func()
plot(kd_31_1c, xlab=expression("m"['i']),ylab="Kernel Density Estimates",
     main="Kernel smoothing of 31 simulated magnitudes", type="p", pch=16, col="blue")
```




```
#Question 1.d
#============
dataFrame <- data.frame();
for(i in 1:200)
{
  kd_1d = density_31_func();
```

```
  dataFrame <- rbind(dataFrame, kd_1d[['y']]);
}

x25 = c()
x975 = c()
for(i in 1:100)
{
  y <- dataFrame[[i]];
  x25 = c(x25, quantile(y,0.025)[[1]])
  x975 = c(x975, quantile(y,0.975)[[1]])
}

plot(c(min(mag_vec),max(mag_vec)),c(0,2), type="n",xlab="magnitude",
     ylab="Kernel density estimation",
     main="2.5th to 97.5th percentile, kernel density estimates")
points(k_density_1b[['x']],k_density_1b[['y']],type="l")
points(k_density_1b[['x']],x25,type="l",col="green")
points(k_density_1b[['x']],x975,type="l",col="blue")

#Question 1.e
#============
latitude <- as.numeric(as.vector(dataMatrixE[,6]))
longitude <- as.numeric(as.vector(dataMatrixE[,7]))
bdw = sqrt(bw.nrd0(latitude)^2+bw.nrd0(longitude)^2) ## a possible default bandwidth
b1 = as.points(longitude,latitude)
bdry = matrix(c(-122,34,-122,38,-118,38,-118,34,-122,34),ncol=2,byrow=T)
z = kernel2d(b1,bdry,bdw)
par(mfrow=c(1,2))
image(z,col=gray((64:20)/64),xlab="longititude",ylab="latitude",main="Map of California")
points(b1)
map('county','California',add=T)
x4 = seq(min(z$z),max(z$z),length=100)
plot(c(0,10),c(.8*min(x4),1.2*max(x4)),type="n",axes=F,xlab="",ylab="")
image(c(- 1:1),x4,matrix(rep(x4,2),ncol=100,byrow=T),add=T,col=gray((64:20)/64))
text(2,min(x4),as.character(signif(min(x4),2)),cex=1)
text(2,(max(x4)+min(x4))/2,as.character(signif((max(x4)+min(x4))/2,2)),cex=1 )
text(2,max(x4),as.character(signif(max(x4),2)),cex=1)
mtext(s=3,l=-3,at=1,"Earthquake magnitude density (points/km^2)")
par(mfrow=c(1,1))


#Question 2.a
#============
dataLA = scan("LAhousingpricesaug2013.txt", skip=1, nlines=269, what="char")
dataMatrixLA = matrix(dataLA,ncol=9,byrow=T)
colNamesLA = scan("LAhousingpricesaug2013.txt", skip=0, nlines=1, what="char")
colnames(dataMatrixLA) = colNamesLA
Y = as.numeric(as.vector(dataMatrixLA[,3]))
X1 = as.numeric(as.vector(dataMatrixLA[,4]))
X2 = as.numeric(as.vector(dataMatrixLA[,7]))
X3 = as.numeric(as.vector(dataMatrixLA[,9]))

notNA = !is.na(Y+X1+X2+X3)
Y = Y[notNA]
X1 = X1[notNA]
X2 = X2[notNA]
X3 = X3[notNA]
```

```
df = data.frame(Y,X1,X2,X3)

#Question 2.b
#============
fit = lm(Y ~ X1 + X2 + X3)
b1 = fit$coef[2]

#Question 2.c
#============
i =1
fit1 <- lm(Y ~ X1 + X2 + X3, data=df[-i,])
b_minus1 = fit1$coef[2]
b = b_minus1 - b1

#Question 2.d
#============
b_all = c()
for(i in 2:217) {
  fit_all <- lm(Y ~ X1 + X2 + X3, data=df[-i,])
  b_minus = fit_all$coef[2]
  b_all[i-1] = b_minus - b1
}

#Question 2.e
#============
b_all = c(b,b_all)
plot(1:217,b_all,type="n",xlab="Values i from 1 to 217",ylab="b1(-i) minus
b1",main="Influences for b1")
points(b_all, col="red")
```