Lisa Over
Homework 13
April 28, 2015

CODE

```
#Convert to dataframe and attach
df = data.frame(faithful)
attach(df)

#Set gibbs parameters

x = eruptions
y = waiting

output = summary(glm(y ~ x))

N = 10000
lag = 50
burnin = 0

#Use intercept and slope coeficients from linear regression on the data as initial a and b parameters
a = output$coef[1,1]
b = output$coef[2,1]

#gibbs function receives seven parameters: "x" and "y" independent and dependent data vectors, "a" and "b" as a and b coefficients for simple linear regression, "N" for number of realizations, "lag" for determining how many realizations to skip between saves, and "burnin" for determining how many realizations to skip before starting to save. Gibbs generates N independent random normal values for a, b, and the variance.

gibbs <- function(x,y,a,b,N,lag,burnin) {

	#obtain length of data
	n = length(x)
	#Set N to be N*lag+burnin
	N <- N*lag + burnin

	#Initialize vectors to hold the alpha, beta, and sigsq realizations
	as = NULL
	bs = NULL
	s2s = NULL

	for(i in 1:N) {

		#Generate a sigsq, s2, based on current alpha, a, and beta, b
```

Lisa Over
Homework 13
April 28, 2015

```
            s2 = 1/rgamma(1, n/2, sum((y-a-b*x)^2)/2)
            a = rnorm(1, sum(y-b*x)/n, sqrt(s2/n))
            b = rnorm(1, sum(x*(y-a))/sum(x^2), sqrt(s2/sum(x^2)))

            #if i is greater than burnin and if i is a multiple of the lag, store alpha,
beta, and siqsq
  if(i > burnin) {
        if(i %% lag == 0) {
            as <- c(as,a)
            bs <- c(bs,b)
            s2s <- c(s2s,s2)
   }
  }
      }

      vectors <- list("alphas" = as, "betas" = bs, "sigsqs" = s2s)
  return(vectors)

}

v = gibbs(x,y,a,b,N,lag,burnin)

mean(v$alphas)
#[1] 33.49228
mean(v$betas)
#[1] 10.72543
mean(v$sigsq)
#[1] 35.22142

alphas = v$alphas
betas = v$betas
sigsqs = v$sigsqs

par(mfrow=c(3,1)) #split plotting window into 3 rows and 1 column
ts.plot(alphas,xlab="Iterations")
hist(alphas,probability=T, cex.lab=1.5, cex.axis=1.5)
acf(alphas,lag.max=500)

par(mfrow=c(3,1)) #split plotting window into 3 rows and 1 column
ts.plot(betas,xlab="Iterations")
hist(betas,probability=T, cex.lab=1.5, cex.axis=1.5)
acf(betas,lag.max=500)

par(mfrow=c(3,1)) #split plotting window into 3 rows and 1 column
```

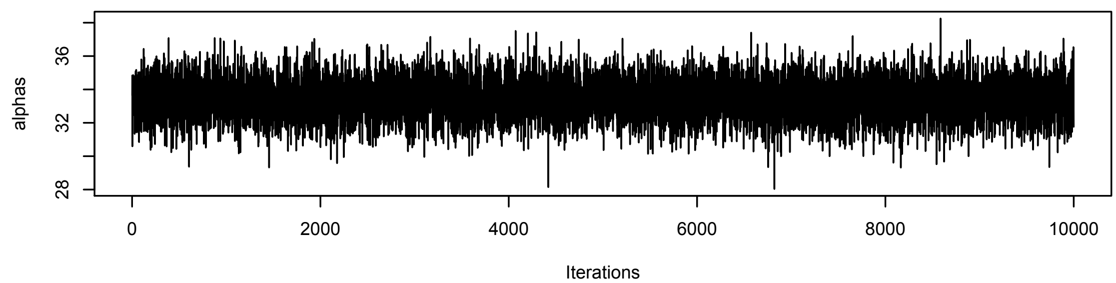
Lisa Over
Homework 13
April 28, 2015

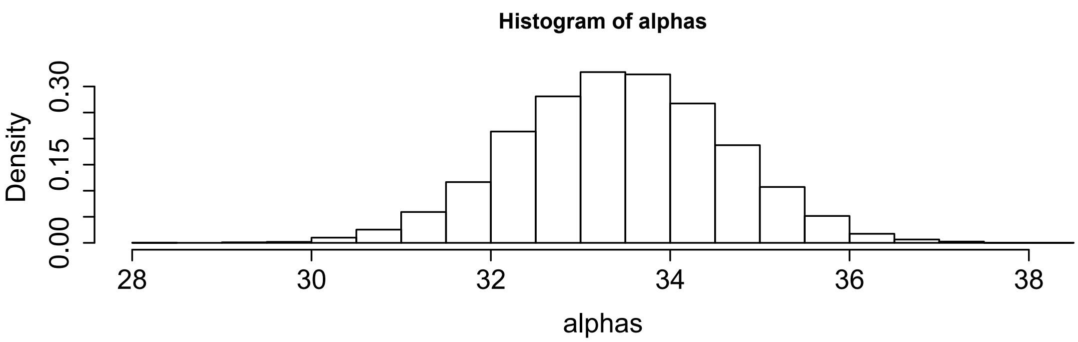
```
ts.plot(sigsqs,xlab="Iterations")
hist(sigsqs,probability=T, cex.lab=1.5, cex.axis=1.5)
acf(sigsqs,lag.max=500)

k=1
mcoef <- matrix(, nrow = N, ncol = 2)
for(i in 1:N) {
		pair = c(alphas[i], betas[i])
		mcoef[k ,] = pair
		k = k + 1
 }

myhat <- matrix(, nrow = N, ncol = length(x))
for(i in 1:N) {
	lines <- mcoef[i,1] + mcoef[i,2]*x
	myhat[i ,] = lines
}

#Calculate the means and the quantiles (0.975 and 0.025) of the columns (2) of the myhat matrix
means = apply(myhat, 2, mean)
q97.5 = apply(myhat, 2, quantile, probs=0.975)
q2.5 = apply(myhat, 2, quantile, probs=0.025)

plot(x,y)
lines(x, means)
lines(x, q97.5)
lines(x, q2.5)
```

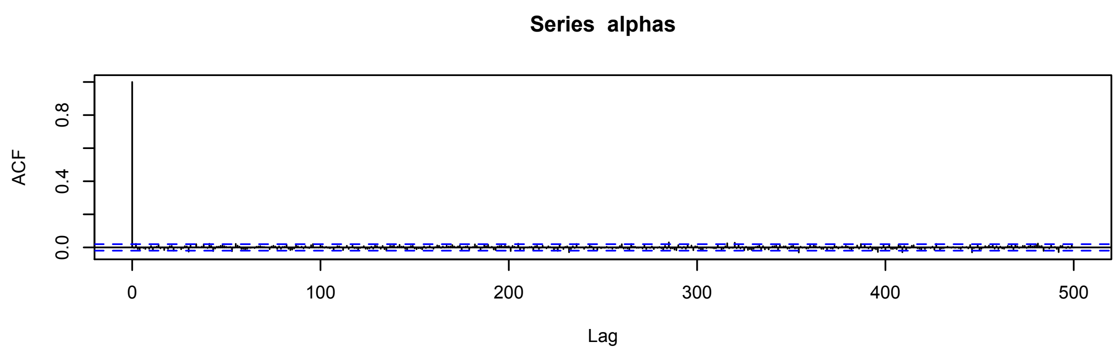
Lisa Over
Homework 13
April 28, 2015

RESULTS

Plots of 10,000 realizations of a, the least-squares regression intercept

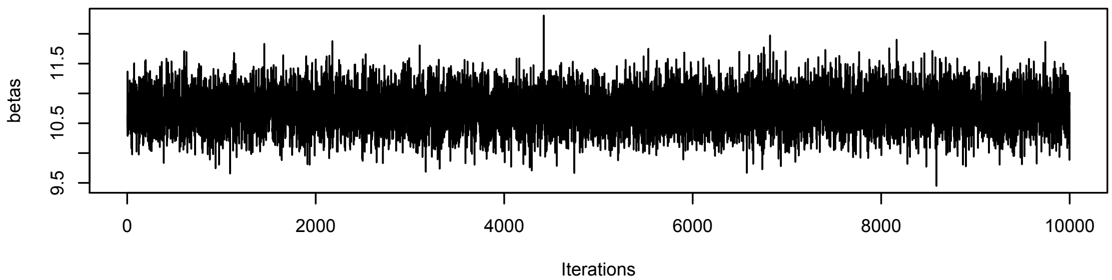


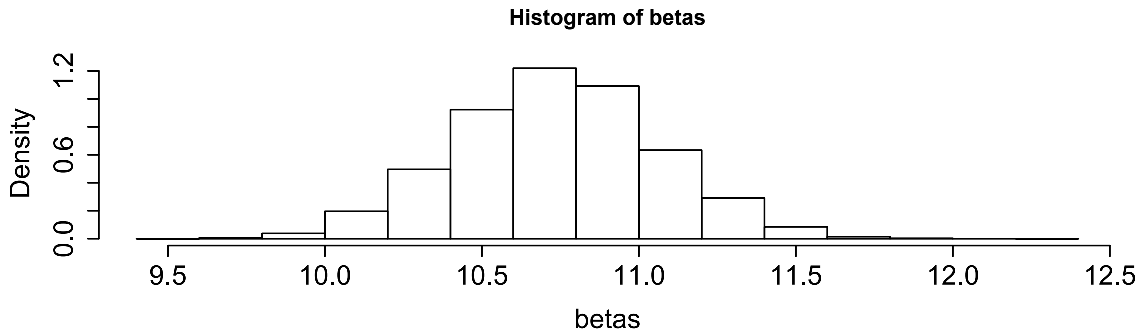


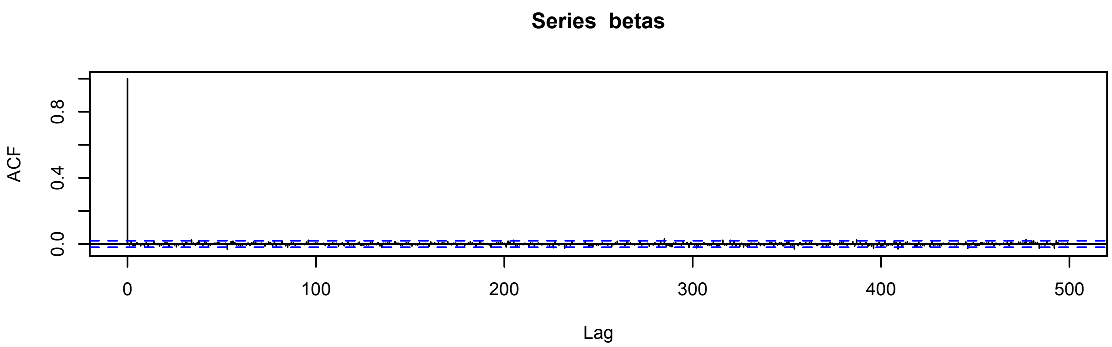


Lisa Over
Homework 13
April 28, 2015

Plots of 10,000 realizations of b, the least-squares regression slope

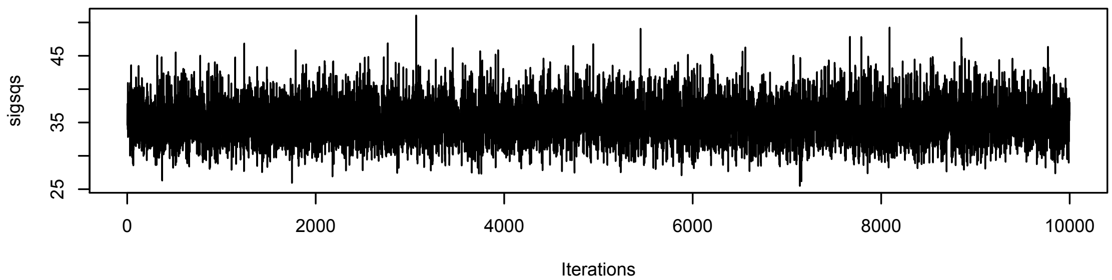


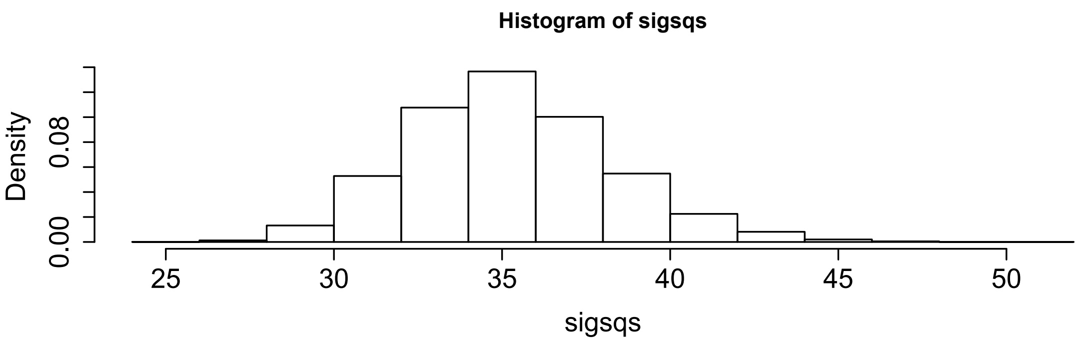


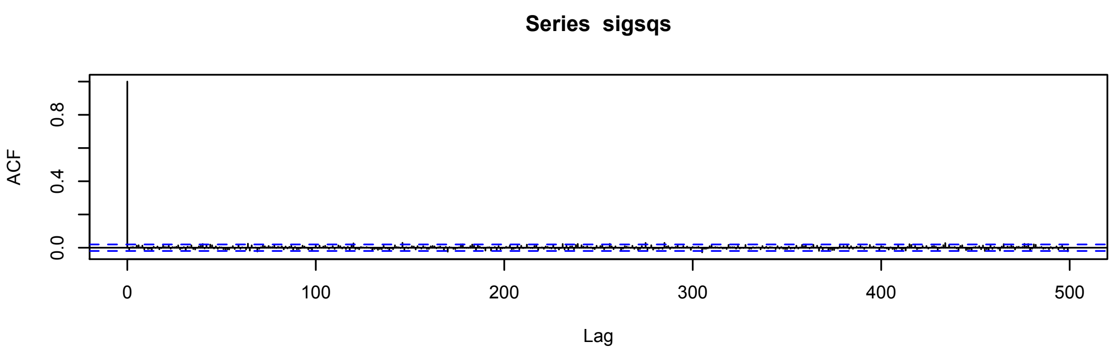


Lisa Over
Homework 13
April 28, 2015

Plots of 10,000 realizations of sigsq, the least-squares regression variance

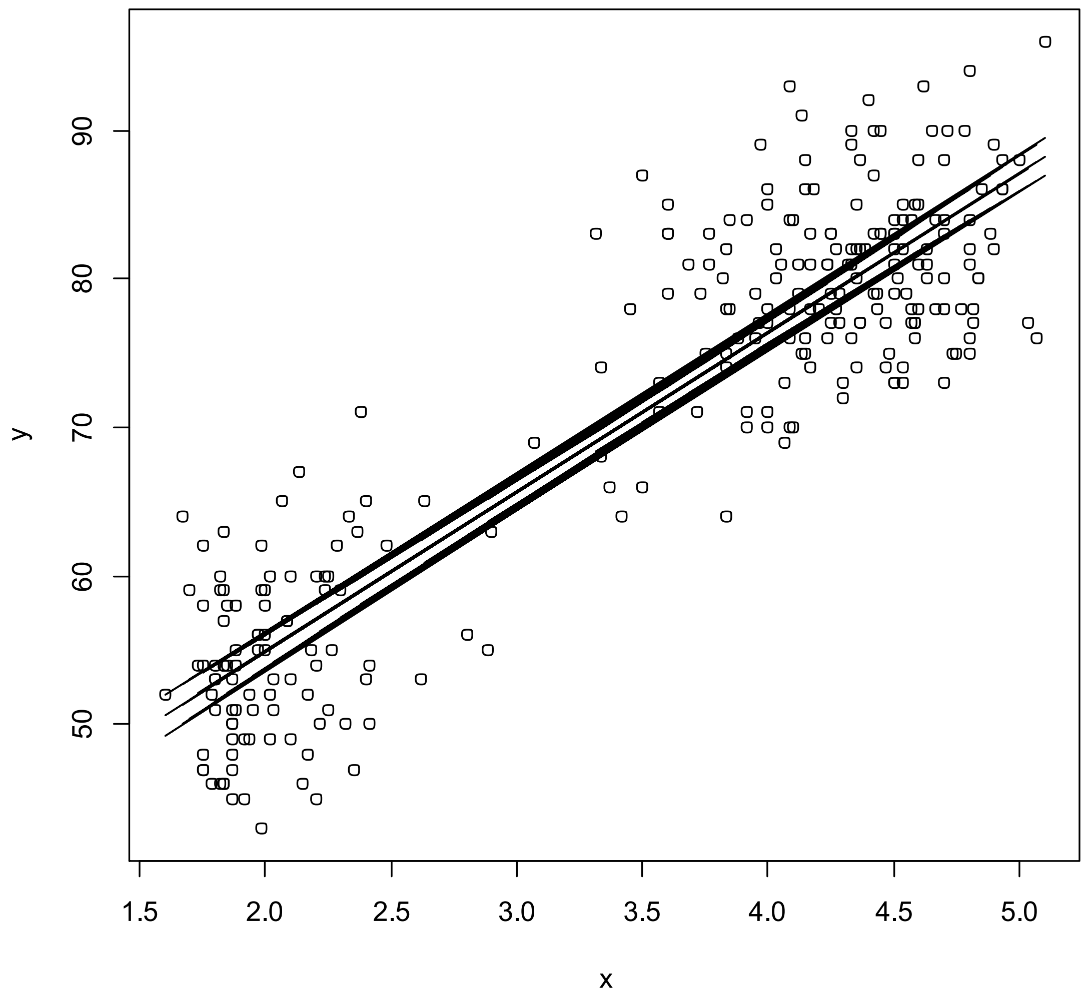






Lisa Over
Homework 13
April 28, 2015

Plot of the data with the estimated least-squares regression line and 95% credible bands



The estimated standard deviation of the regression is 5.93 minutes and the estimated least squares line is

$\hat{y}$ = 33.5 + 10.7x,

where x is the duration of the eruption in minutes and $\hat{y}$ is the predicted waiting time between eruptions in minutes.