Homework 7

Jing Leng (GSI: Jiahe)

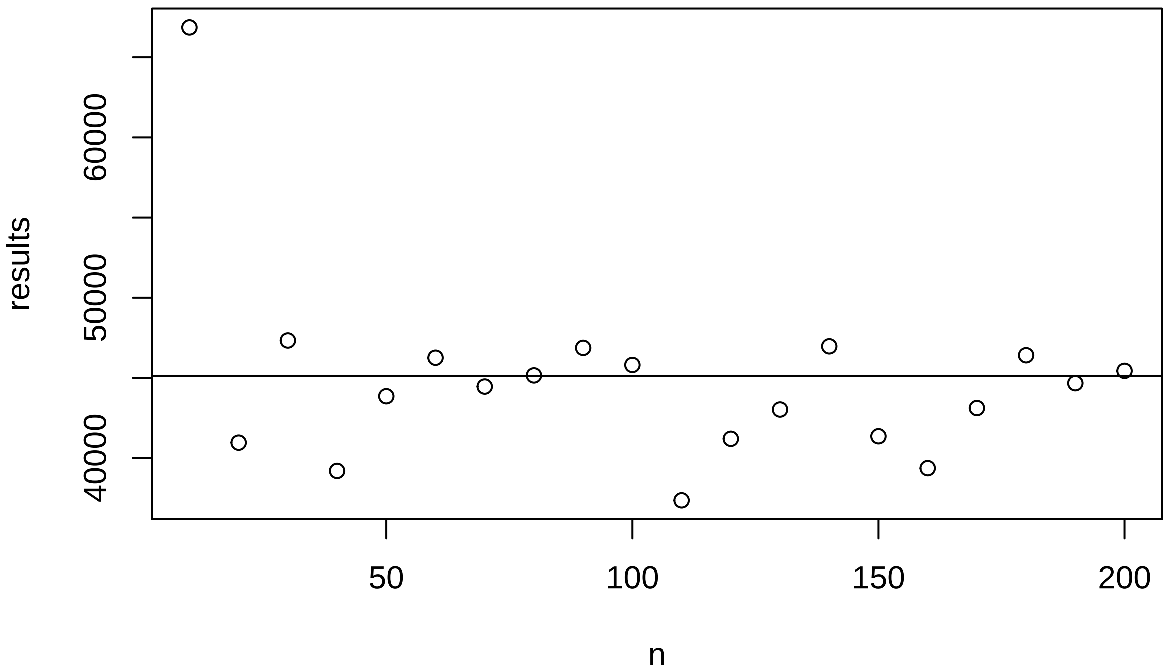
November 10, 2014

1

```
LAIdata = read.table("LAI.csv", sep = ",")
LAIdata <- as.matrix(LAIdata)
LAI <- sapply(1:72, function(i)matrix(LAIdata[i,], nrow = 120, ncol = 60, byrow = T),simplify =
                  "array")
LAI2005 <- LAI[,,1:12]
truesum <- sum(LAI2005)

avgLAI = apply(LAI, c(1, 2), mean)
avgL = (avgLAI > 0) * 3 # land is now uniformly colored
length(which(avgL != 0)) # here, 2443 entries indicate land
```

```
## [1] 2443
```

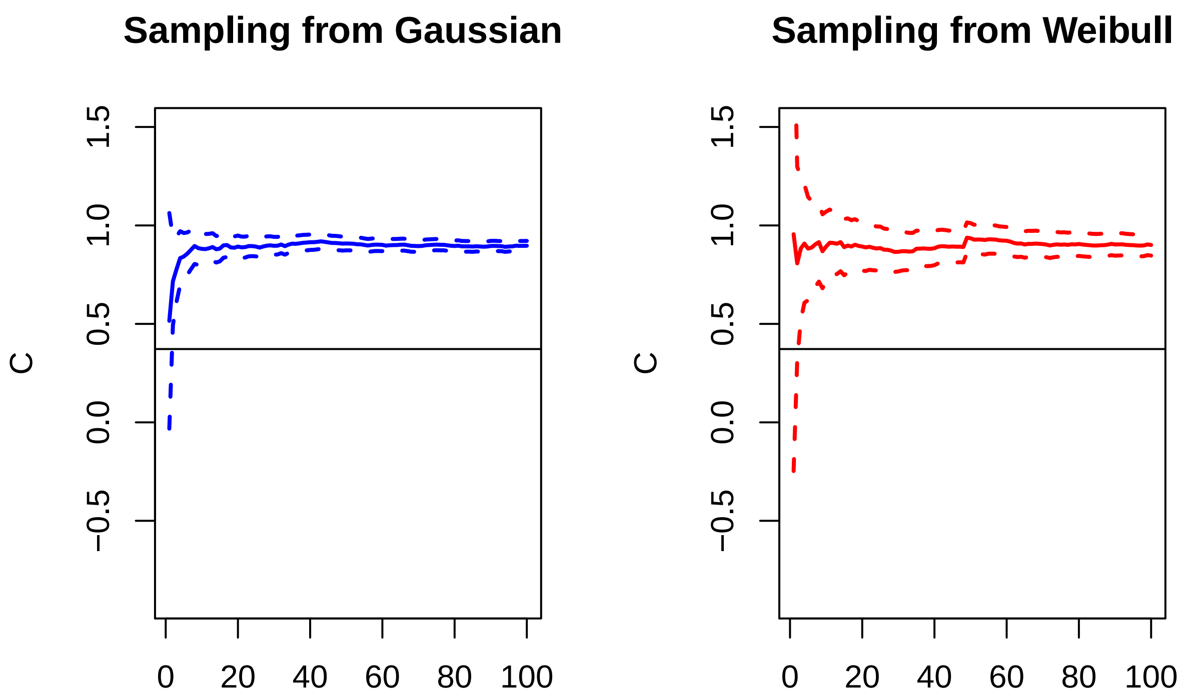
```
land.ind = which(avgL != 0, arr.ind = T)
land.mat = as.matrix(land.ind) # 2443 x 2 matrix that has indices of land
ind = seq(1, 2443, by = 1) # row numbers of land.mat
n = seq(10,200, 10) # number of locations (change it to whatever we like: 5, 10, ... <= 2443)

mc <- function(n) {
  land.loc = sample(ind, size = n, replace = F)
  samp <- array(0, dim(LAI)[1:2])
  for(i in 1:n){
    samp[land.mat[land.loc[i], 1], land.mat[land.loc[i],2]] = sum(LAI2005[land.mat[land.loc[i], 1], lan
  }
  est <- sum(samp)*2443/n
  return (est)
}

results <- sapply(n, mc)
plot(n, results)
abline(h = truesum)
```



2

```
## define the h function for the Gaussian case
hgauss = function(x){
  h_x = sqrt(pi/2)*(sin(x))^2*(x^2)
    return(h_x)
}

## define the h function for the Weibull case
hweibull = function(x){
    h_x = 0.5*(sin(x))^2*x*exp(x^2/2)
}

## sample path
sample_seq = seq(20,10000,by=100)

## generate the data first
X_gauss = rnorm(max(sample_seq))
X_weibull = rweibull(max(sample_seq),shape=2,scale=1) #k=2,lambda=1#
h_gauss = hgauss(X_gauss)
h_weibull = hweibull(X_weibull)

# initialize a list of two components to save the result
Res = list()
Res$gauss = array(0,c(length(sample_seq),3))
Res$weibull = array(0,c(length(sample_seq),3))

z = qnorm(0.975)  # upper 0.025 quantile

## calculate and plot
```

```
for (i in 1:length(sample_seq)){
    subspl_gauss = h_gauss[1:sample_seq[i]]
    subspl_weibull = h_weibull[1:sample_seq[i]]

    m_gauss = mean(subspl_gauss)
    m_weibull = mean(subspl_weibull)

    se_gauss = sqrt(var(h_gauss)/sample_seq[i])
    se_weibull = sqrt(var(h_weibull)/sample_seq[i])

    # save in a row, mean, lower bd, upper bd
    Res$gauss[i,] = c(m_gauss,m_gauss-z*se_gauss,m_gauss+z*se_gauss)
    Res$weibull[i,] = c(m_weibull,m_weibull-z*se_weibull,m_weibull+z*se_weibull)
}

C_true = 0.5*(1-3/exp(1)^2)*sqrt(pi/2)
## now plot
par(mfrow=c(1,2))
plot(Res$gauss[,1],type="l",lwd=2,ylab="C",xlab="",ylim=c(-0.9,1.5),col=4)
lines(Res$gauss[,2],lwd=2,lty=2,col=4)
lines(Res$gauss[,3],lwd=2,lty=2,col=4)
abline(h=C_true)
title("Sampling from Gaussian ")
plot(Res$weibull[,1],type='l',lwd=2,col=2,ylab="C",xlab="",ylim=c(-0.9,1.5))
lines(Res$weibull[,2],col=2,lwd=2,lty=2)
lines(Res$weibull[,3],col=2,lwd=2,lty=2)
abline(h=C_true)
title("Sampling from Weibull")
```




```
## part 3
# this is omega(x) function
```

```
omegafunc = function(x){
    return(2/0.372*(sin(x))^2*exp(-x^2/2+x))
}
Y = rgamma(max(sample_seq),shape=3,rate=1)
Yminus1sq = (Y-1)^2
Y_ind = (1<Y)*(Y<5)
omega_Y = omegafunc(Y)

ResI = array(0,c(length(sample_seq),4))

for (i in 1:length(sample_seq)){
    subspl = Y_ind[1:sample_seq[i]]*omega_Y[1:sample_seq[i]]*Yminus1sq[1:sample_seq[i]]
    m = mean(subspl)
    MC_error = sqrt(var(subspl)/sample_seq[i])
    lb = m-MC_error*z
    ub = m+MC_error*z
    ResI[i,] = c(m,lb,ub,MC_error)
}
plot(ResI[,1],type="l",lwd=2,ylim=c(min(ResI[,-4]),max(ResI[,-4])),xlab="",ylab="I")
lines(ResI[,2],col=2,lwd=2,lty=2)
lines(ResI[,3],col=2,lwd=2,lty=2)
title("Importance Sampling from Gamma(3,1)")
```