

HW5 - STAT 580 - Sp 2015

Yet T Nguyen

03/26/2015

1.

```
(a) #1.
#a)
set.seed(1)
p <- .3
lambda <- 2
n <- 100
r <- rbinom(n, 1, p)
y <- rpois(n, lambda)
x <- r*y
x

##      [1] 0 0 0 6 0 1 0 0 0 0 0 0 0 0 1 0 3 0 0 2 6 0 0 0 0 0 0 0 1 0 0 0 0 0 4
##     [36] 0 2 0 6 0 3 0 1 0 0 2 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1
##     [71] 0 3 0 0 0 4 2 0 2 4 0 1 0 0 4 0 3 0 0 0 0 0 0 0 4 1 2 0 0 1 0
```

```
(c) #b)
a<- 1;b <- 1
gibbs <- function(a,b){
  r <- rep(1, n)
  cnt <- 0
  burn.in <- 10000
  keep <- 0
  ns <- 10000
  pkeep <- lambdakeep <- NULL
  repeat{
    lambda <- rgamma(1,a +sum(x), b +sum(r))
    p <- rbeta(1,1+sum(r), n+1-sum(r))
    pnnew <- p*exp(-lambda)/(p*exp(-lambda) + (1-p)*(x==0))
    r <- rbinom(n, 1, pnnew)
    cnt <- cnt +1
    if (cnt >burn.in){
      keep <- keep +1
      pkeep[keep] <- p
      lambdakeep[keep] <- lambda
    }
    if (cnt >burn.in + ns) break
  }

  cip <- quantile(pkeep, prob = c(.025, .975))
  cilambda <- quantile(lambdakeep, prob = c(.025, .975))
  res <- rbind(p=cip, lambda=cilambda)
  res
}

gibbs(1,1) # the confidence intervals do contain the true values
```

```
##           2.5%      97.5%
## p       0.2355416 0.4386853
## lambda 1.6742864 2.8436237

gibbs(1,2) # the confidence intervals do contain the true values

##           2.5%      97.5%
## p       0.2387191 0.4471769
## lambda 1.6022768 2.7320347

gibbs(1000,1000) # the confidence intervals do not contain the true values

##           2.5%      97.5%
## p       0.3290509 0.6016109
## lambda 0.9682580 1.0897239

gibbs(2, 1000) # the confidence intervals do not contain the true values

##           2.5%      97.5%
## p       0.86273583 0.99907441
## lambda 0.05486017 0.08580084
```

2.

```
#2.
theta1 <- 1.5
theta2 <- 2

f <- function(z, theta1 = 1.5, theta2 = 2){
  z^(-3/2)*exp(-theta1*z-theta2/z +2*sqrt(theta1*theta2) + log(sqrt(2*theta2)))
}

mh <- function(a, b){
  i <- 1
  x <- NULL
  x[1] <- 1
  repeat{
    y <- rgamma(1, a, b)
    gy <- dgamma(y, a, b)
    r <- f(y)*dgamma(x[i], a, b)/(f(x[i])*dgamma(y, a, b))
    u <- runif(1,0,1)
    x[i+1] <- ifelse(u < r, y, x[i])
    i <- i+1
    if (i == 10000) break
  }
  # hist( x, nclass = 100, prob = T)
  abs(c(mx.diff = mean(x), m1x.diff = mean(1/x))-c(mz = sqrt(theta2/theta1),
                                                    m1z=sqrt(theta1/theta2) + 1/(2*theta2)))
}

mh(1,.1)

##      mx.diff      m1x.diff
## 0.008498937 0.015303931

mh(1,1) # better
```

```
##      mx.diff      m1x.diff
## 0.0008119234 0.0078215705

mh(2,2)

##      mx.diff      m1x.diff
## 0.006390013 0.013114352
```