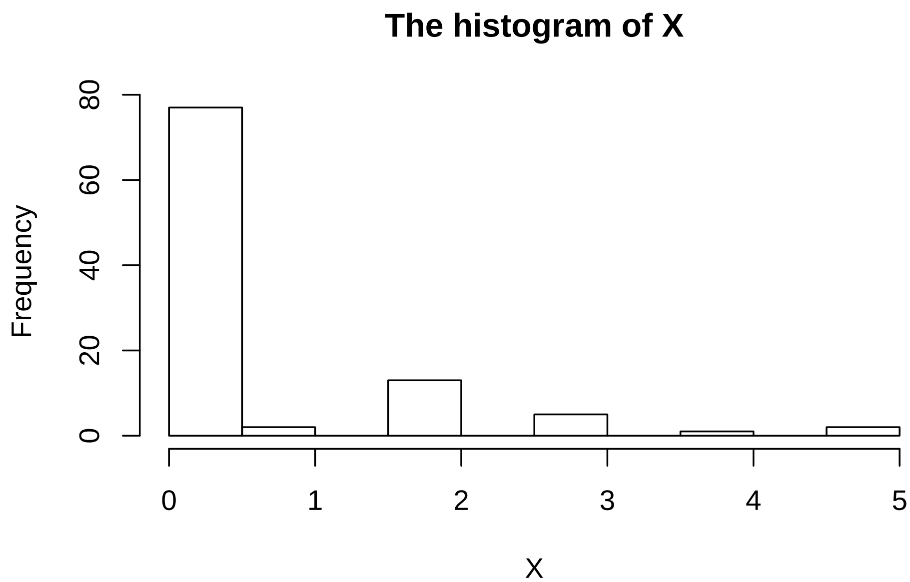
Problem 1

(a)




```r
p <- 0.3
lambda <- 2
n <- 100
set.seed(580)
Y <- rpois(n, lambda)
R <- rbinom(n, 1, p)
X <- R*Y
```

(b)

$$
\begin{aligned}
f(\lambda|p,\boldsymbol{r},\boldsymbol{x}) &\propto \lambda^{a-1+\sum_{i=1}^{n} x_i} \exp(-b\lambda + \lambda \sum_{i=1}^{n} x_i) \\
f(p|\lambda,\boldsymbol{r},\boldsymbol{x}) &\propto p^{\sum_{i=1} nr_i}(1-p)^{n-\sum_{i=1} nr_i} \\
f(r_i|\lambda,p,\boldsymbol{x}) &\propto \left(e^{-\lambda}\frac{p}{1-p}\right)^{r_i} r_i^{x_i}
\end{aligned}
$$

(c)

- For $a = 1$ and $b = 1$, the 95% confidence interval of λ is $(1.511976, 2.824471)$, and the 95% confidence interval of p is $(0.1802548, 0.3769580)$.

Problem 1 continued on next page...

- For $a = 1$ and $b = 10$, the 95% confidence interval of λ is $(1.013360, 1.918225)$, and the 95% confidence interval of p is $(0.2049541, 0.4450217)$.
- For $a = 10$ and $b = 1$, the 95% confidence interval of λ is $(1.921057, 3.296289)$, and the 95% confidence interval of p is $(0.1799374, 0.3731253)$.
- For $a = 10$ and $b = 10$, the 95% confidence interval of λ is $(1.306673, 2.288253)$, and the 95% confidence interval of p is $(0.1927605, 0.3973745)$.

```
Gibbs_Sampler <- function(m, X, Y, a, b,init_lambda, init_p, init_r){
  n <- length(X)
  lambda <- rep(NA,m)
  p <- rep(NA,m)
  r <- matrix(NA, n, m)
  lambda[1] <- init_lambda
  p[1] <- init_p
  r[,1] <- init_r
  for(k in 2:m){
    lambda[k] <- rgamma(1,a+sum(X), b+sum(r[,k-1]))
    p[k] <- rbeta(1,1+sum(r[,k-1]),n+1-sum(r[,k-1]))
    r[,k] <- sapply(1:n, function(t){rbinom(1,1,p[k]*exp(-lambda[k])/
                                          (p[k]*exp(-lambda[k])+(1-p[k]))
                                          *ifelse(X[t]==0,1,0))})
  }
  return(list(lambda, p, r))
}

gibsamp_1_1 <- Gibbs_Sampler(1000,X,Y,1,1,2,0.3,R)
quantile(gibsamp_1_1[[1]], c(0.025,0.975))
quantile(gibsamp_1_1[[2]], c(0.025,0.975))
```

Problem 2

- For $Gamma(1, 1)$, the absolute error of $E(Z)$ is 0.0106 and the absolute error of $E(1/Z)$ is 0.0081.
- For $Gamma(1, 10)$, the absolute error of $E(Z)$ is 0.2968 and the absolute error of $E(1/Z)$ is 0.1942.
- For $Gamma(10, 1)$, the absolute error of $E(Z)$ is 0.4547 and the absolute error of $E(1/Z)$ is 0.3125.

- For $Gamma(10, 10)$, the absolute error of $E(Z)$ is 0.0289 and the absolute error of $E(1/Z)$ is 0.0117.
- For $Gamma(0.1, 0.1)$, the absolute error of $E(Z)$ is 0.0096 and the absolute error of $E(1/Z)$ is 0.0134.

R code:

```r
theta_1 <- 1.5
theta_2 <- 2
Expec_Z <- sqrt(theta_2/theta_1)
Expec_Inv_Z <- sqrt(theta_1/theta_2) + 1/(2*theta_2)
f <- function(z){
  z^{-1.5}*exp(-theta_1*z-theta_2/z+2*sqrt(theta_1*theta_2)+
                  log(sqrt(2*theta_2)))
}

IMH <- function(m, gamma_shape, gamma_rate, init){
  X <- rep(NA, m)
  X[1] <- init
  for(i in 2:m){
    U <- runif(1,0,1)
    Y <- rgamma(1,gamma_shape, gamma_rate)
    r <- min(f(Y)/f(X[i-1])*dgamma(X[i-1],gamma_shape, gamma_rate)/
              dgamma(Y,gamma_shape, gamma_rate),1)
    if(U<=r) X[i] <- Y
    else X[i] <- X[i-1]
  }
  return(X)
}

IMH_sample <- IMH(10000,0.1,0.1,0.7)
abs(mean(IMH_sample) - Expec_Z)
abs(mean(1/IMH_sample) - Expec_Inv_Z)
```

Problem 3

C code:

```c
#include <R.h>
#include <Rinternals.h>
#include <Rmath.h>
```

```
SEXP Gibbs_Sampler(SEXP Rm, SEXP RX, SEXP RY, SEXP Ra,
            SEXP Rb, SEXP Rinit_lambda,
            SEXP Rinit_p, SEXP Rinit_r){
        int m = asInteger(Rm), *init_r = INTEGER(Rinit_r), n, k, t;
        double a = asReal(Ra), b = asReal(Rb);
        double init_lambda = asReal(Rinit_lambda);
        doubleinit_p = asReal(Rinit_p);
        int *X = INTEGER(RX), *Y = INTEGER(RY);
        int sum_X = 0, sum_r_k=0;

        n = length(RX);

        SEXP lambda = PROTECT(allocVector(REALSXP, m));
        SEXP p = PROTECT(allocVector(REALSXP, m));
        SEXP r = PROTECT(allocMatrix(INTSXP, m, n));
        SEXP Rout = PROTECT(allocVector(REALSXP, 2*m));

        REAL(lambda)[0] = init_lambda;
        REAL(p)[0] = init_p;
        for(k=0;k<n;k++){
                INTEGER(r)[k] = init_r[k];
                sum_X += X[k];
        }

        GetRNGstate();
        for(k=1;k<m;k++){
                sum_r_k = 0;
                for(t=0;t<n;t++){
                        sum_r_k +=INTEGER(r)[(k-1)*n+t];
                }

                REAL(lambda)[k] = rgamma(a+sum_X, 1.0/(b+sum_r_k));
                REAL(p)[k] = rbeta(1+sum_r_k,n+1-sum_r_k);
                for(t=0;t<n;t++){
                        if(X[t]==0)
                            INTEGER(r)[k*n+t] = rbinom(1,
                                        REAL(p)[k]*exp(-REAL(lambda)[k])/
                                        (REAL(p)[k]*exp(-REAL(lambda)[k])+
                                        1 - REAL(p)[k]));
                         else INTEGER(r)[k*n+t] = 1;
```

```
            }

    }
    PutRNGstate();

    for(k=1;k<2*m;k++){
            if(k<m) REAL(Rout)[k] = REAL(lambda)[k];
            else REAL(Rout)[k] = REAL(p)[k-m];
    }

    UNPROTECT(4);
    return(Rout);
}
```