PS4

Hanyu Wang

1. What is a crude estimate of the secondary attack rate (SAR) among the 41 households with at least one case, assuming the incubation period is 2-14 days and the maximum serial interval is 21 days? (For now, assume there is only 1 index case per household.)

```
# Number of contacts ranges 1 to 10 (columns)
contacts <- 1:7

# Number of contacts infected ranges 0 to 7 (rows)
infect <- 0:7

colnames(PS3covid$final.size) <- paste0("Num_Contact ",contacts)
rownames(PS3covid$final.size) <- paste0("Num_Infected ",infect)

# Number of households observed in which "infect"=i (rows) out of "contacts"=m
# (columns) members were infected
PS3covid$final.size
```

```
##                Num_Contact 1 Num_Contact 2 Num_Contact 3 Num_Contact 4
## Num_Infected 0             95            52            44            30
## Num_Infected 1              6             3             2             1
## Num_Infected 2              0             2             2             0
## Num_Infected 3              0             0             3             1
## Num_Infected 4              0             0             0             2
## Num_Infected 5              0             0             0             0
## Num_Infected 6              0             0             0             0
## Num_Infected 7              0             0             0             0
##                Num_Contact 5 Num_Contact 6 Num_Contact 7
## Num_Infected 0             25            22            13
## Num_Infected 1              2             2             3
## Num_Infected 2              1             1             2
## Num_Infected 3              2             0             1
## Num_Infected 4              2             0             0
## Num_Infected 5              0             1             1
## Num_Infected 6              0             0             0
## Num_Infected 7              0             0             1
```

$$\hat{SAR} = \frac{number\ of\ persons\ exposed\ who\ develop\ disease}{total\ number\ of\ susceptible\ persons\ exposed}$$

```
infected_households <- colSums(PS3covid$final.size[2:8, ])
total_households <- sum(infected_households)

infected_counts <- 0:7
```

```r
total_infected_by_size <- colSums(PS3covid$final.size * infected_counts)

# Numerator of SAR (How many individuals were infected?)
secondary_cases <- sum(total_infected_by_size) - total_households

# Denominator of SAR (How many individuals were at risk (susceptible)?)
exposed_contacts <- sum(infected_households * ((1:7) - 1))

# Thus, an estimated SAR is:
p.est <- secondary_cases / exposed_contacts
p.est
```

```
## [1] 0.3779528
```

Answer:

The crude estimate of the secondary attack rate (SAR) among the 41 households with at least one case is 0.3779528.

First, assume that we ONLY observed the final number of household members infected for each of the 322 households ("final.size(i+1, j)" = number of households of size j in which i household members were infected, and "final_size(1, j)" = number of households of size j in which nobody was infected).

2. Use the method of Longini et al (1982) to estimate the probability of being infected from the point source/outside the household (i.e. the CPI) and from an infected household member (i.e. the SAR).

Probability j out of k household members infected over the course of the epidemic:

$$m_{jk} = \binom{k}{j} m_{jj} B^{(k-j)} Q^{j(k-j)} \text{ for } j < k$$

$$m_{kk} = 1 - \sum_{j=0}^{k-1} m_{jk}$$

Assume that there are n households in total, where a_{jk} = the number of HHs that had j of k susceptible members infected. Calculate the log-likelihood given B and Q of all the observed number of households with j out of k family members infected ("data(j+1,k)"=a_{jk}). THen use *optim()* to minimize the negative log-likelihood.

$$L(Q, B) = \prod_{k=1}^{K} \prod_{j=0}^{k} (m_{jk})^{a_{jk}}$$

$$log\ L = C + \sum_{k} \sum_{j} a_{jk}\ (log\ m_{jj} + (k-j)\ log\ B + j\ (k-j)\ log\ Q)$$

```r
# Negative log-likelihood of data given B and Q
source(file="/Users/macbook/Desktop/longiniLL.R")

BQest <- optim(c(0.9,0.7),longiniLL,data=PS3covid$final.size)$par
BQest
```

```
## [1] 0.9544426 0.7821942
```

The probability of being infected from the point source/outside the household:

$$CPI = 1 - \hat{B}$$

The probability of being infected from an infected household member:

$$SAR = 1 - \hat{Q}$$

```
CPI <- 1 - BQest[1]
CPI
```

```
## [1] 0.04555735
```

```
SAR <- 1 - BQest[2]
SAR
```

```
## [1] 0.2178058
```

Answer:

The probability of being infected from the point source/outside the household (i.e. the CPI) is 0.04556998, and that from an infected household member (i.e. the SAR) is 0.2177673.

3. Assuming the infectious period is 5 days, what is the probability of transmission per day given your estimate of Q (i.e. the probability of escaping infection from an infected household member over the entire course of their illness)? (For this question, assume that infectiousness does not vary over the course of infection.)

Probability of escaping infection on each day of infectious period, q_i, can be estimated by using $\hat{Q} = \prod_{i=0}^{i_{max}} q_i$, where $i_{max} = 5$.

Since the infectiousness (p) does not vary over the course of infection, the probability of escaping infection ($q = 1 - p$) does not vary.

The infectious period is 5 days, then

$$\hat{Q} = q^5 = (1-p)^5 \longrightarrow p = 1 - Q^{\frac{1}{5}}$$

.

```
D <- 5
Qhat <- BQest[2]

p_day <- 1 - Qhat^(1/D)
p_day
```

```
## [1] 0.04794307
```

Answer:

The probability of transmission per day given the estimate of Q is 0.04794307.

4. Suppose there was also an ongoing influenza epidemic in the community, and some influenza cases were misclassified as COVID-19 cases. Would this lead to over- or underestimation of the SAR (based on the method of Longini et al)? Explain your answer.

Answer:

Misclassifying influenza cases as COVID-19 would result in an increase in *the number of infection from the community* in the data, so the estimated probability of escaping infection from the community, $\hat{B}$, would be underestimated.

$\hat{CPI} = 1 - \hat{B}$, CPI would be overestimated.

However, since the probability j out of k household members infected over the course of the epidemic, $m_{jk} = \binom{k}{j} m_{jj} B^{(k-j)} Q^{j(k-j)}$ for $j < k$, was fixed, the underestimated of B would result in a overestimation of Q.

Therefore, $\hat{SAR} = 1 - \hat{Q}$ would be underestimated.

Next, assume that we observed all infection and recovery events as they occurred each hour during the 3-week follow-up period in the 41 households with at least 1 case. The matrices S, E, I, and R represent the number of susceptible, exposed/ latently infected, infectious, and recovered people, respectively, at hour i (rows) in household j (columns). Assume the risk of infection from outside the household is negligible. Thus, the likelihood is:

$$L(S, I, t_{infect}) = \prod_{j=1}^{41} \left(\prod_{i=1}^{504} e^{-\beta S_{ij} I_{ij} (t_{i+1} - t_i)} \right)^{S_{504,j}} \left(\beta S_{t_{infect} j} I_{t_{infect} j} \prod_{i=1}^{t_{infect}-1} e^{-\beta S_{ij} I_{ij} (t_{i+1} - t_i)} \right)^{S_{1,j} - S_{504,j}}$$

where t_{infect} (="$tinfect(:,j)$") is the time (in hours) at which infection events occurred in household j (and ="NaN" otherwise).

5. Write down the equation for the log-likelihood.

$$\ell(\beta) = \sum_{j=1}^{41} \left(S_{504,j} \sum_{i=1}^{504} \left(-\beta S_{ij} I_{ij} (t_{i+1} - t_i) \right) + (S_{1,j} - S_{504,j}) \left(\log \left(\beta S_{t_{\text{infect}},j} I_{t_{\text{infect}},j} \right) + \sum_{i=1}^{t_{\text{infect}}-1} \left(-\beta S_{ij} I_{ij} (t_{i+1} - t_i) \right) \right) \right)$$

$\ell(\beta) = -\beta \sum_{j=1}^{41} \left(S_{504,j} \sum_{i=1}^{504} S_{ij} I_{ij} (t_{i+1} - t_i) + (S_{1,j} - S_{504,j}) \sum_{i=1}^{t_{infect}-1} S_{ij} I_{ij} (t_{i+1} - t_i) \right) + \sum_{j=1}^{41} (S_{1,j} - S_{504,j}) \log \left(\beta, S_{t_{infect},j} I_{t_{infect},j} \right)$

6. Estimate β via MCMC using a random walk on the log scale. (HINT: Use the code "hhsirLL" from lab 7 to calculate the log-likelihood; make sure you output the positive log-likelihood rather than the negative log-likelihood.)

Plot a histogram of the posterior distribution and report the median and 95% credible interval of your estimate.

```
# Equation can be found in a Word document for Lab 7 and Slide 30 in Lecture 7

hhsirLL = function(beta,S,I,tinfect){ # "beta" is a parameter we want to estimate. S, I, & tinfect are
  hh = dim(S)[2]   # number of households   = number of columns in the matrix S (50 HHs in Lab 7)
  tmax = dim(S)[1] # duration of follow-up  = number of rows in the matrix S (504 hours in Lab 7)
  dt = 1/24 # time step (in days, i.e. one-24th day=hour, because the questions asks for the rate of tr

  S0 = S[1,] # Initial number of susceptibles in each household
  Sf = S[tmax,] # Final number of susceptibles in each household
  ninfect = S0 - Sf # number infected in each household

  logLL = matrix(0,tmax,hh)
```

```
  for (j in 1:hh){ # j represents each household
    for (t in 1:tmax){ #  i in equation

      ##### First chunk of the equation in Lab 7 #####

      # log likelihood contribution of uninfected people in household j
      logLL[t,j] = logLL[t,j] - Sf[j]*beta*S[t,j]*I[t,j]*dt
    }

    ##### Second chunk of the equation in Lab 7 #####

    if (ninfect[j]>0){ # if the household has at least one case
      for (k in 1:ninfect[j]){ # for each infected person (k) in household j
        if (tinfect[k,j]>0){
          for (t in 1:(tinfect[k,j]-1)){
            # log-likelihood for escaping prior to time of infection
            logLL[t,j] = logLL[t,j] - beta*S[t,j]*I[t,j]*dt
          }
          # log likelihood they did NOT escape at time of infection
          # (Because we are using discrete time SIR model, we have to use CDF
          #  rather than PDF.)
          logLL[t,j] = logLL[t,j] + log(
            1-exp(-beta*S[tinfect[k,j],j]*I[tinfect[k,j],j]*dt))
        }
      }
    }
  }
  return(sum(logLL))
}
```

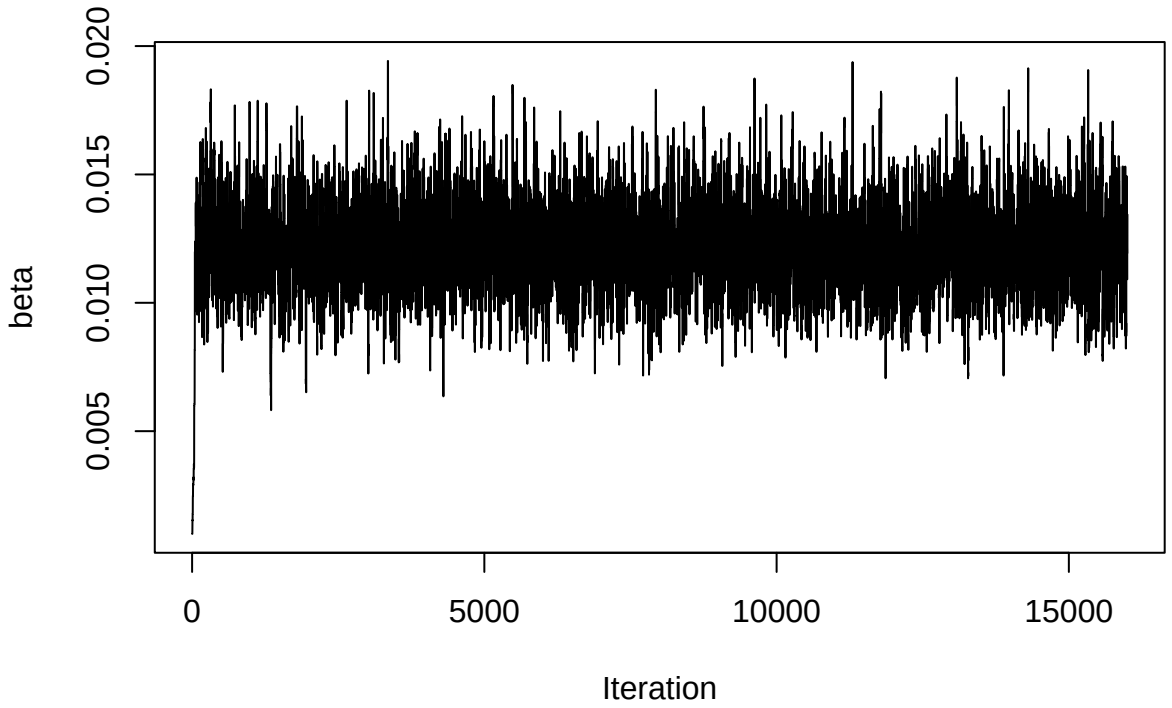
```
betaHHest <- optimize(f=hhsirLL, S=PS3covid$S, I=PS3covid$I,
                      tinfect=PS3covid$tinfect, interval=c(0,10), maximum=T)
betaHHest$maximum
```

```
## [1] 0.01268293
```

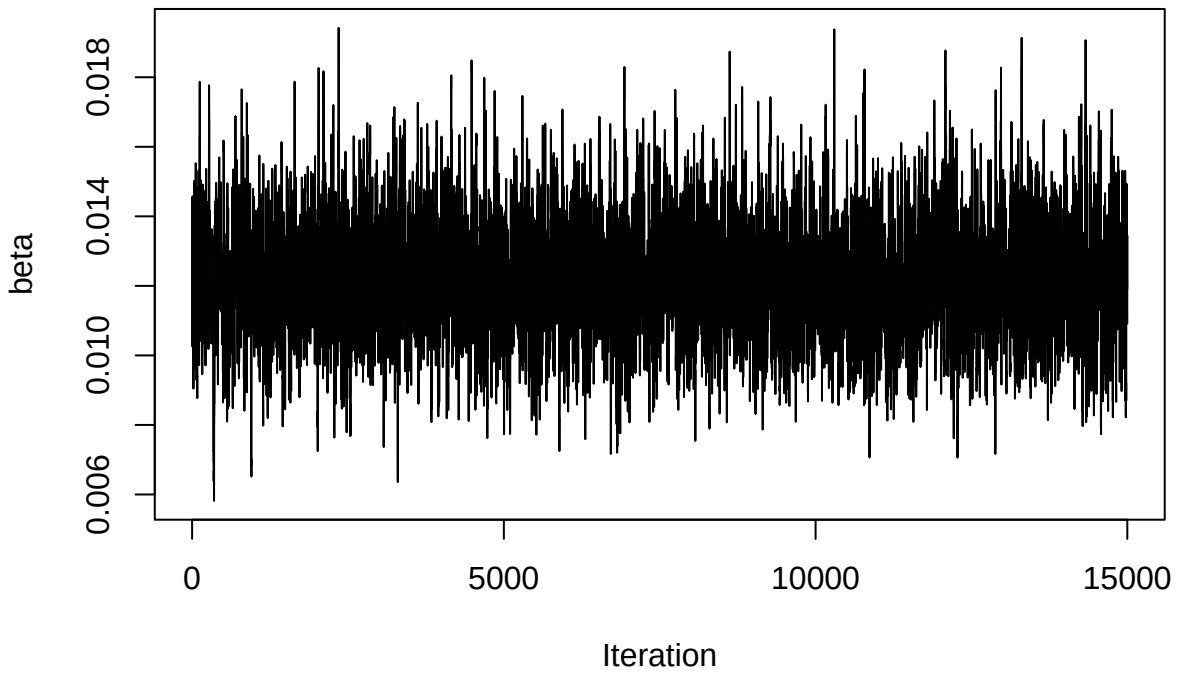
```
# Make an empty vector to store sampled parameters in
sampled_beta <- c()

# Set the walk rate of the MCMC sampler,
#i.e. how big are the jumps going to be
steprate <- 0.2

# Give a starting value for beta (arbitrary choice)
current_beta <- 0.001

# Number of steps to run the sampler
burn_in_steps <- 1000
run_steps     <- 15000
total_steps   <- burn_in_steps + run_steps

# Get a starting log_likelihood value
current_log_LL <- hhsirLL(current_beta, S=PS3covid$S, I=PS3covid$I, tinfect=PS3covid$tinfect)
```

```r
# Implement MCMC
# Hint: Slide 32-34 from Lecture 7

#------------checking log-normal distribution------------
# stepRate <- 0.5
# currentbeta <- 0.25
# vec_rnorm <- rnorm(5000)
# vec_beta_cand <- currentbeta*exp(stepRate*vec_rnorm)
# hist(vec_beta_cand,100)
# hist(log(vec_beta_cand),100)
# #repeat with greater values of stepRate, what do you notice?
# rm(stepRate,currentbeta,vec_rnorm,vec_beta_cand)

#------------Okay now back to lab..------------
for (i in 1:total_steps) {
  # 1. Propose a new value for beta. We use a
  # random walk on the log scale to ensure that the parameter
  # value stays positive, since hazards < 0 don't make sense
  candidate_beta  <- current_beta*exp(steprate*rnorm(1))

  # 2. The proposal ratio allows us to correct for the fact that
  # the proposal distribution (above) is asymmetric
  log_proposal_ratio <- dlnorm(candidate_beta/current_beta,
           meanlog = 0, sdlog = 1, log = T) -
    dlnorm(current_beta/candidate_beta,
                               meanlog = 0, sdlog = 1, log = T)

  # 3a. Get the log-likelihood for the newly sampled parameter
  candidate_log_LL <- hhsirLL(candidate_beta, S=PS3covid$S, I=PS3covid$I, tinfect=PS3covid$tinfect)

  # 3b. Log-likelihood ratio for last and current sampled parameters
  log_likelihood_ratio <- candidate_log_LL - current_log_LL

  # 4. Get the acceptance probability by exponentiating
  acceptance_prob <- exp(log_proposal_ratio + log_likelihood_ratio)

  # 5. Draw a random number from (0, 1] and if it's less than
  # the acceptance probability, accept the candidate parameter.
  # Otherwise, stay on the current one
  if (runif(1) < acceptance_prob) {
    current_beta <- candidate_beta
    current_log_LL <- candidate_log_LL
  }
  # Save the current parameter to the trace of sampled parameters
  sampled_beta <- append(sampled_beta, current_beta)
  }



#-------------------------------------------------------------------------------#
# Examine MCMC outputs
#-------------------------------------------------------------------------------#
```

```r
# Trace plots for beta (including burn in period)
sampled_beta_w_burnin  <- sampled_beta[1:total_steps]
plot(sampled_beta_w_burnin,type='l',xlab='Iteration',ylab='beta')
```

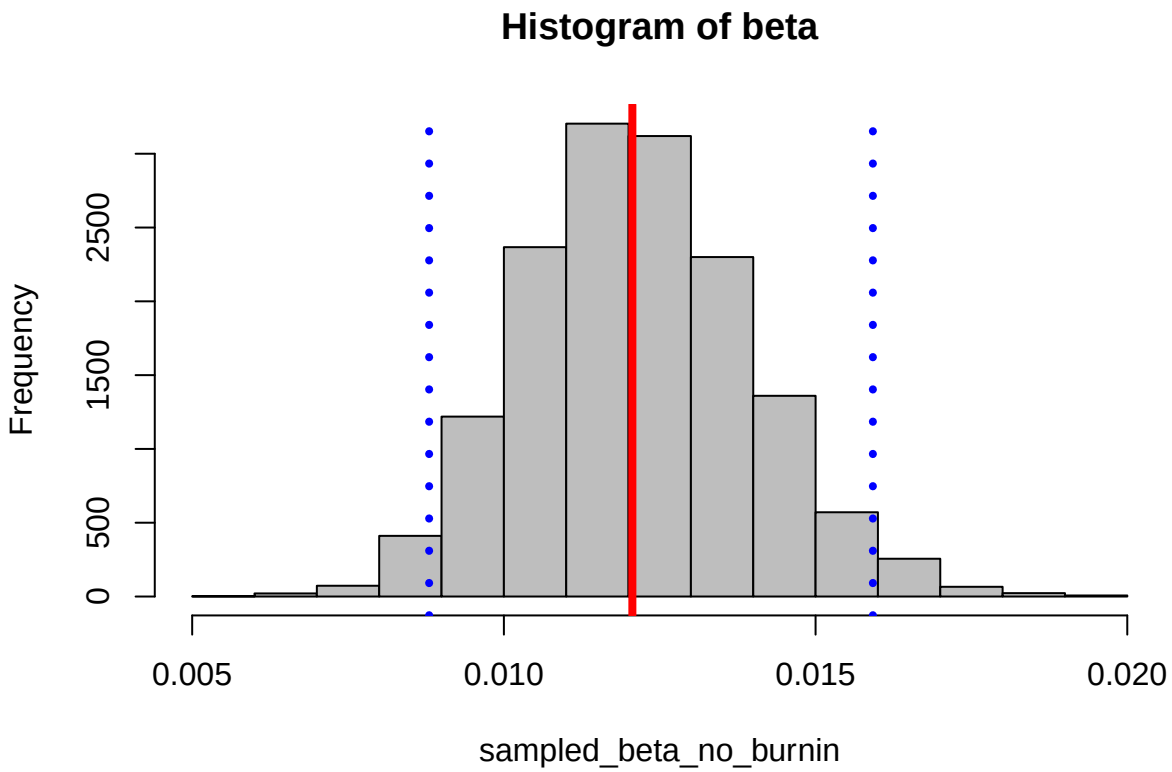



```r
# Trace plots for beta (excluding burn in period)
sampled_beta_no_burnin <- sampled_beta[(burn_in_steps+1):total_steps]
plot(sampled_beta_no_burnin,type='l',xlab='Iteration',ylab='beta')
```




```r
# Posterior distribution for beta (excluding burn in period)
hist(sampled_beta_no_burnin, col='grey', main='Histogram of beta')
```

```r
abline(v=median(sampled_beta_no_burnin),lwd=4,col='red') # Median
abline(v=quantile(sampled_beta_no_burnin,
                  c(0.025,0.975)),lwd=4,col='blue',lty=3) # 95% credible interval
```




```r
# What is the estimate of beta and the corresponding
# 95% Bayesian credible intervals? (*Question 1 in Lab 7)
quantile(sampled_beta,c(0.025,0.5,0.975))
```

```
##        2.5%         50%       97.5%
## 0.008722889 0.012058670 0.015928255
```

Answer:

The trace plots for β show stable oscillation, and the posterior distribution demonstrates a roughly symmetric, unimodal distribution, indicating that the Markov chain has converged well and mixed properly.

The estimated β is 0.01268293. The median of estimate is 0.012058670. The 95% credible interval is (0.008722889,0.015928255).

EXTRA CREDIT: What if we assume that the risk of infection among household members not infected during the point source outbreak was constant over time (i.e. exponential survival)? The vector "event.times" gives the time of symptom onset (in hours) of all non-index cases, and is equal to 505 hours (i.e. T+1) for all household members who did not exhibit symptoms during the 3-week follow-up period.

7. Estimate the constant hazard rate, λ, via MCMC and given the median and 95% credible interval of your estimate in 1/days. (HINT: Modify the code for "simple_survival" from lab 7.)

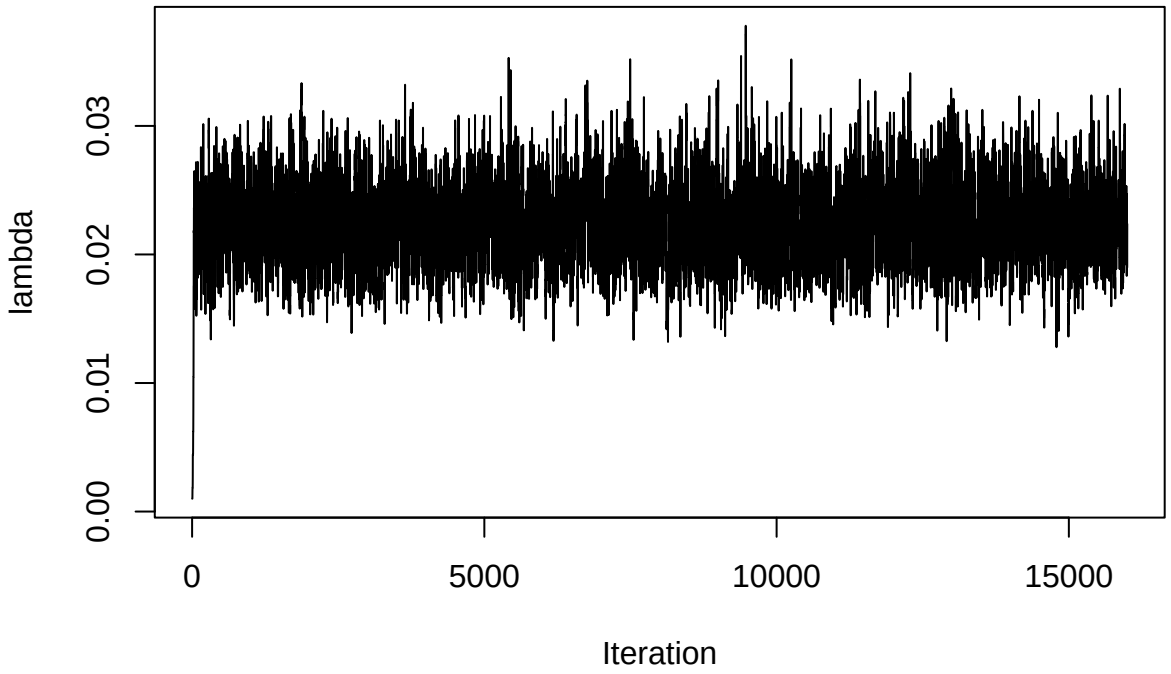
```
# Simulate data
event_times <- PS3covid$event.times
t <- 505-1
```

```
# Define a log-likelihood function for this censored dataset
# Hint: Slide 31 from Lecture 7
exp_survival_LL <- function(event_times, lambda_per_day, t) {
  lambda_per_hour <- lambda_per_day / 24 #####

  # Create an empty vector to store results
  log_LL <- 0
  # Step 1. Calculate the log-likelihood for non-censored events (PDF)
  #         NOTE: This is the first equation on Slide XX
  log_LL <- log_LL + sum(dexp(event_times[event_times <= t],
                              rate = lambda_per_hour, log = TRUE))

  # Step 2.Add the log-likelihood for censored individuals (1 - CDF)
  # NOTE: This is the second equation on Slide XX

  # NOTE: CDF for exponential distribution = 1 - exp(-lambda*t).
  #       Thus, exp(-lambda*t) on Slide XX is 1 - CDF.
  # NOTE: CDF is the prob that people got infected by Time t, and thus,
  #       1 - CDF is the prob that people escaped infection until Time t.
  # NOTE: 1 - CDF for exponential distrib. can be calculated using pexp()
  #       by setting "lower = FALSE". If lower = TRUE, it will calculate
  #       CDF.
  log_LL <- log_LL + sum(pexp(event_times[event_times > t],
                              rate = lambda_per_hour, lower = FALSE, log = TRUE))
  return(log_LL)
}
```

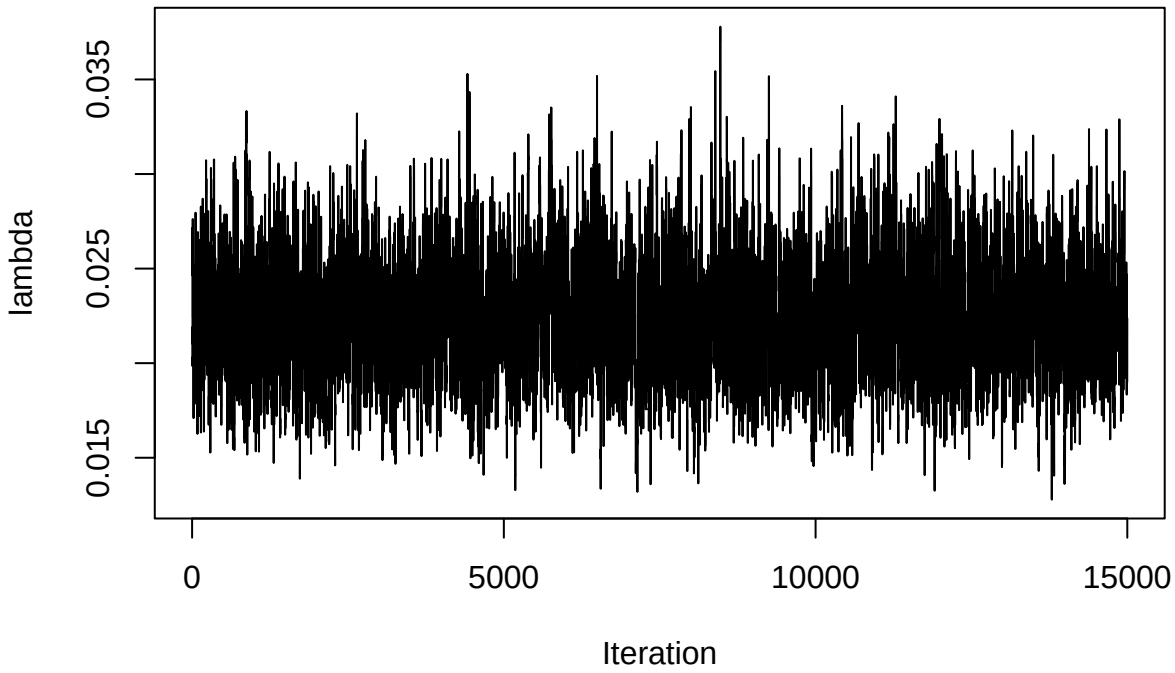
```
# Make an empty vector to store sampled parameters in
sampled_lambda <- c()

# Set tLhe walk rate of the MCMC sampler,
#i.e. how big are the jumps going to be
steprate <- 0.2

# Give a starting value for lambda (arbitrary choice)
current_lambda <- 0.001

# Number of steps to run the sampler
burn_in_steps <- 1000
```

```r
run_steps     <- 15000
total_steps   <- burn_in_steps + run_steps

# Get a starting log_likelihood value
current_log_LL <- exp_survival_LL(event_times, current_lambda, t)

# Implement MCMC
# Hint: Slide 32-34 from Lecture 7

#------------checking log-normal distribution------------
# stepRate <- 0.5
# currentLambda <- 0.25
# vec_rnorm <- rnorm(5000)
# vec_lambda_cand <- currentLambda*exp(stepRate*vec_rnorm)
# hist(vec_lambda_cand,100)
# hist(log(vec_lambda_cand),100)
# #repeat with greater values of stepRate, what do you notice?
# rm(stepRate,currentLambda,vec_rnorm,vec_lambda_cand)

#------------Okay now back to lab..------------
for (i in 1:total_steps) {
  # 1. Propose a new value for lambda. We use a
  # random walk on the log scale to ensure that the parameter
  # value stays positive, since hazards < 0 don't make sense
  candidate_lambda  <- current_lambda*exp(steprate*rnorm(1))

  # 2. The proposal ratio allows us to correct for the fact that
  # the proposal distribution (above) is asymmetric
  log_proposal_ratio <- dlnorm(candidate_lambda/current_lambda,
           meanlog = 0, sdlog = 1, log = T) -
    dlnorm(current_lambda/candidate_lambda,
                               meanlog = 0, sdlog = 1, log = T)

  # 3a. Get the log-likelihood for the newly sampled parameter
  candidate_log_LL <- exp_survival_LL(event_times, candidate_lambda, t)

  # 3b. Log-likelihood ratio for last and current sampled parameters
  log_likelihood_ratio <- candidate_log_LL - current_log_LL

  # 4. Get the acceptance probability by exponentiating
  acceptance_prob <- exp(log_proposal_ratio + log_likelihood_ratio)

  # 5. Draw a random number from (0, 1] and if it's less than
  # the acceptance probability, accept the candidate parameter.
  # Otherwise, stay on the current one
  if (runif(1) < acceptance_prob) {
    current_lambda <- candidate_lambda
    current_log_LL <- candidate_log_LL
  }
  # Save the current parameter to the trace of sampled parameters
  sampled_lambda <- append(sampled_lambda, current_lambda)
  }
```

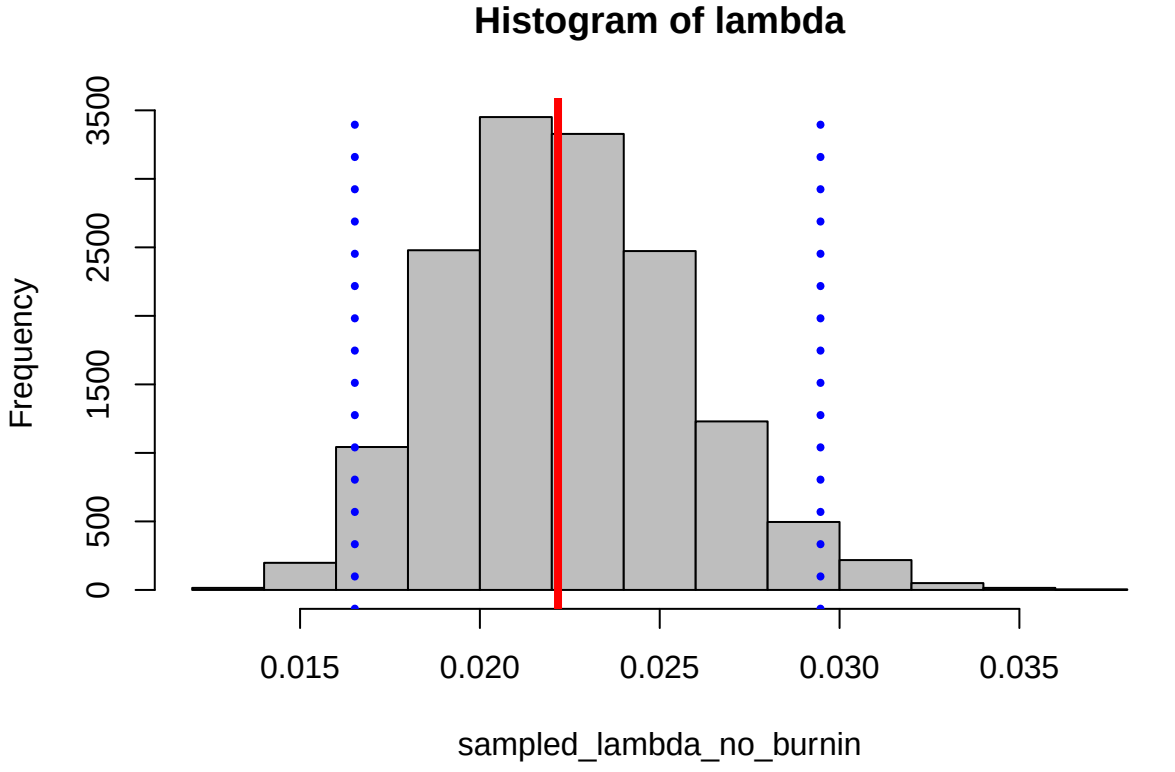
```r
#-------------------------------------------------------------------------------#
# Examine MCMC outputs
#-------------------------------------------------------------------------------#

# Trace plots for lambda (including burn in period)
sampled_lambda_w_burnin  <- sampled_lambda[1:total_steps]
plot(sampled_lambda_w_burnin,type='l',xlab='Iteration',ylab='lambda')
```




```r
# Trace plots for lambda (excluding burn in period)
sampled_lambda_no_burnin <- sampled_lambda[(burn_in_steps+1):total_steps]
plot(sampled_lambda_no_burnin,type='l',xlab='Iteration',ylab='lambda')
```




```
# Posterior distribution for lambda (excluding burn in period)
hist(sampled_lambda_no_burnin, col='grey', main='Histogram of lambda')
abline(v=median(sampled_lambda_no_burnin),lwd=4,col='red') # Median
abline(v=quantile(sampled_lambda_no_burnin,
                  c(0.025,0.975)),lwd=4,col='blue',lty=3) # 95% credible interval
```




```
# What is the estimate of lambda and the corresponding
# 95% Bayesian credible intervals? (*Question 1 in Lab 7)
```

```r
quantile(sampled_lambda,c(0.025,0.5,0.975))
```

```
##       2.5%        50%      97.5%
## 0.01648051 0.02214409 0.02939156
```

```r
# How does it compare to the lambda used to generate the data?
```

Answer:

The trace plots for λ show stable oscillation, and the posterior distribution demonstrates a roughly symmetric, unimodal distribution, indicating that the Markov chain has converged well and mixed properly.

The median of estimate in 1/days is 0.02214409. The 95% credible interval in 1/days is (0.01648051, 0.02939156).