Assignment 2. undergrad level.

Problem 1.

$$\theta = E[Y/X] = E[\exp(9 + 2\log X + \varepsilon)]$$

$$= E[E(X^2 e^{9+\varepsilon} | X)]$$

$$= E[X^2 \cdot E(e^{9+\varepsilon} | X)]$$

$$= E(X^2 \cdot e^9 \cdot E(e^{\varepsilon} | X)]$$

$\varepsilon \sim N(0,1)$ $\quad E(e^{\varepsilon}) = M_{\varepsilon}(1)$ ← MGF of Normal.

$= e^{\frac{1}{2}}$

$$= e^{9.5} \cdot E(X^2)$$

RB : $\hat{\theta}_{RB} = e^{9.5} E(X^2) = \frac{\sum_{i=1}^{n} X_i^2}{n} \cdot e^{9.5}$ $\quad X_i \sim \text{lognormal}(0,1)$ $\quad$ use $n = 1000 \times 50$.

$= 100029$

$Var(\hat{\theta}_{RB}) = 527338298$

MC : $\hat{\theta}_{MC} = E[Y/X]$

$= \frac{1}{n} \sum_{i=1}^{n} \exp(9 + 2\log X + \varepsilon)$ $\quad X \sim \text{lognorm}(0,1)$ $\quad n = 1000 \times 50$.

$\varepsilon \sim N(0,1)$

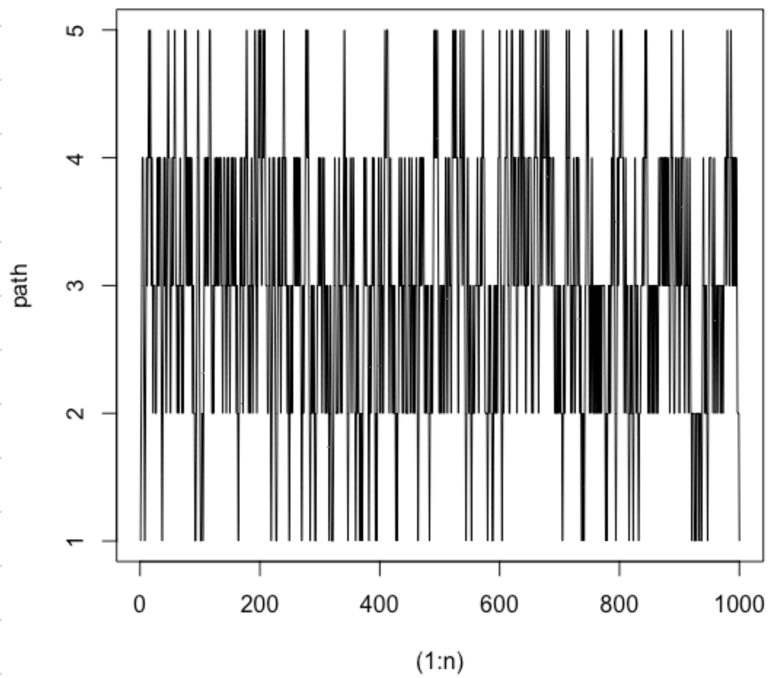
$= 104793$

$Var(\hat{\theta}_{MC}) = 1042630387$.

We can see the RB estimator performs better than MC estimator, as it has lower variance.

Problem 2.

a). time vs state



b). relative freq: $\hat{\pi} = ($ mean(path==1), ..., mean(path==5))

$= (0.058, 0.255, 0.372, 0.255, 0.060)$

I guess stationary dist^{n} π is $\hat{\pi}$

c) $P\hat{\pi} = (0.0626, 0.2462, 0.3804, 0.2478, 0.063)$

$\approx \hat{\pi}$

theoretically. $\underset{\sim}{\pi} P = \underset{\sim}{\pi}$.

$$\Rightarrow \begin{cases} 0.2\pi_1 + 0.2\pi_2 = \pi_1 & \Rightarrow \pi_1 = \frac{1}{4}\pi_2. \\ 0.8\pi_1 + 0.2\pi_2 + 0.4\pi_3 = \pi_2 & \Rightarrow \pi_2 = \frac{2}{3}\pi_3 \\ 0.6\pi_2 + 0.2\pi_3 + 0.6\pi_4 = \pi_3 & \Rightarrow \pi_3 = \frac{3}{2}\pi_4. \\ 0.4\pi_3 + 0.2\pi_4 + 0.8\pi_5 = \pi_4 & \Rightarrow \pi_4 = 4\pi_5 \\ 0.2\pi_4 + 0.2\pi_5 = \pi_5. & \Rightarrow \pi_5 = \pi_5. \\ \pi_1 + \pi_2 + \pi_3 + \pi_4 + \pi_5 = 1 \end{cases}$$

$\Rightarrow \pi = (0.0625, 0.25, 0.375, 0.25, 0.0625)$

which is close to $\hat{\pi}$

thus stationary dist^{n} $\pi = \hat{\pi}$ As desired.

Problem 3

a) posterior

$$g(\alpha, \eta \mid x_1, \cdots, x_n) \propto f(x_1, \cdots, x_n \mid \alpha, \eta) \cdot \pi(\alpha, \eta)$$

$$\propto \prod_{i=1}^{n} \alpha \, \eta^{\beta} \, x_i^{\alpha-1} \, e^{-\eta x_i^{\alpha} - \alpha - c\eta}$$

$$\propto \prod_{i=1}^{n} \alpha \, \eta^{\beta} \, x_i^{\alpha-1} \, e^{-\eta(x_i^{\alpha}+c)-\alpha}$$

$$\propto \alpha \, \eta^{\beta} \left(\prod_{i=1}^{n} x_i^{\alpha-1}\right) e^{-\eta\left(\prod_{i=1}^{n} x_i^{\alpha}+c\right)-\alpha}.$$

b) acceptance prob:

$$\rho(\alpha', \eta' \mid \alpha^{(t)}, \eta^{(t)}) = \min\left\{ \frac{g(\alpha', \eta' \mid X_{1:n}) \, q(\alpha^{(t)}, \eta^{(t)} \mid \alpha', \eta')}{g(\alpha^{(t)}, \eta^{(t)} \mid X_{1:n}) \, q(\alpha', \eta' \mid \alpha^{(t)}, \eta^{(t)})}, 1 \right\}$$

$$= \min\left\{ \frac{\alpha' \eta'^{\beta} \left(\prod_{i=1}^{n} x_i^{\alpha'-1}\right) e^{-\eta'\left(\prod_{i=1}^{n} x_i^{\alpha'}+c\right)-\alpha'} \cdot \frac{1}{\alpha' \eta'} e^{-\frac{\alpha^{(t)}}{\alpha'} - \frac{\eta^{(t)}}{\eta'}}}{\alpha^{(t)} \eta^{(t)\beta} \left(\prod_{i=1}^{n} x_i^{\alpha^{(t)}-1}\right) e^{-\eta^{(t)}\left(\prod_{i=1}^{n} x_i^{\alpha^{(t)}}+c\right)-\alpha^{(t)}} \cdot \frac{1}{\alpha^{(t)} \eta^{(t)}} e^{-\frac{\alpha'}{\alpha^{(t)}} - \frac{\eta'}{\eta^{(t)}}}}, 1 \right\}$$

c) chain is generated by MH Algo:

① init $n=0$ and $\alpha^{(n)}, \eta^{(n)} = 1$

② sample α_new, η_new from $\exp(\alpha^{(n)})$ and $\exp(\eta^{(n)})$ $u \sim U(0,1)$

③ if $u \leq e$ ← accepting rate by b).

$\alpha^{(n+1)} = \alpha$_new, $\eta^{(n+1)} = \eta$_new.

④ goto ②. n++.

```r
beta = 0.1
c = 0.5

g = function(a, b) {
    x = c(0.56, 2.26, 1.90, 0.94, 1.40, 1.39, 1.00, 1.45, 2.32, 2.08, 0.89, 1.68)
    return(a*b^beta*prod(x^(a-1))*exp(-b*(prod(x^a)+c)-a))
}


n = 10001
path_a = vector("numeric", n)
path_b = vector("numeric", n)
path_a[1] = 1
path_b[1] = 1

for (i in 2:n){
    a = path_a[i-1]
    b = path_b[i-1]
    u = runif(1)
    a_new = rexp(1, a)
    b_new = rexp(1, b)
    rate = g(a_new, b_new)*dexp(a, a_new)*dexp(b, b_new)/g(a, b)/dexp(a_new, a)/dexp(b_new, b)
    if (u <= rate){
        path_a[i] = a_new
        path_b[i] = b_new
    } else {
        path_a[i] = a
        path_b[i] = b
    }
}
```

d). mean (path_a) = 0.816.

mean (path_b) = 0.095.

e). 95% CI by R.

```
CI_alpha = quantile(path_a[2000:10001], c(0.025, 0.975))
CI_ite = quantile(path_b[2000:10001], c(0.025, 0.975))
```

CI_alpha = (0.248, 1.313)

CI_ita = (0.00623, 0.46263)

Problem 4.

a). $f(x|y) \propto x^2 e^{-xy^2} e^{-y^2} e^{2y} e^{-4x}$

$\propto x^2 e^{-x(4+y^2)} \quad \sim \text{Gamma}(3, 4+y^2)$

$g(y|x) \propto x^2 e^{-xy^2} e^{-y^2} e^{2y} e^{-4x}$

$\propto e^{(-xy^2 - y^2 + 2y)}$

$\propto e^{-(y^2(x+1) - 2y)}$

$\propto e^{-(y^2 - 2\frac{y}{(x+1)} + \frac{1}{(x+1)^2} - \frac{1}{(x+1)^2}) \cdot (x+1)}$

$\propto \exp\left[\left(-(y - \frac{1}{x+1})^2 - \frac{1}{(x+1)^2}\right) \cdot (x+1)\right]$

$\propto \exp\left(-(y - \frac{1}{x+1})^2 \cdot \frac{1}{x+1}\right)$

$\propto \exp\left(-\frac{(y - \frac{1}{x+1})^2}{2 \cdot (\frac{1}{2(x+1)})}\right) \quad \sim N\left(\frac{1}{x+1}, \frac{1}{2(x+1)}\right)$

the full conditional of π follow Gamma and Normal

b) ① Init $n=0$. $X_n=1$, $Y_n=1$

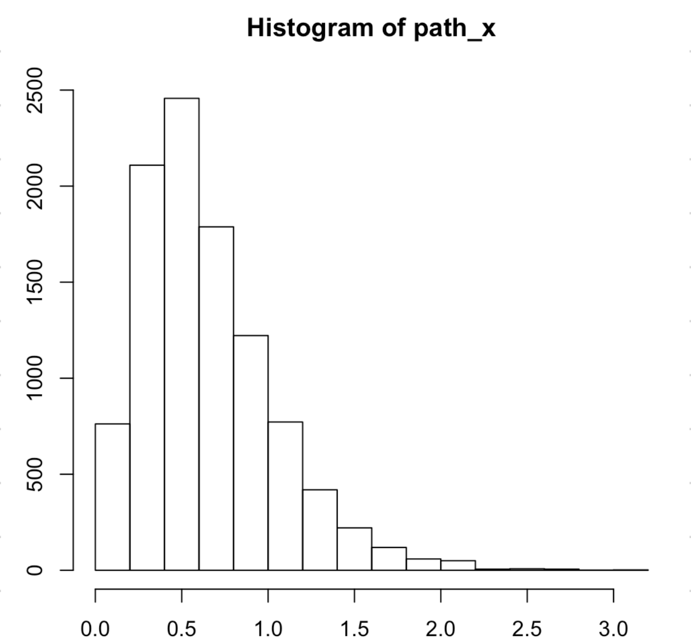
② Generate $X_{n+1} \sim Gamma(3, 4+Y_n^2)$

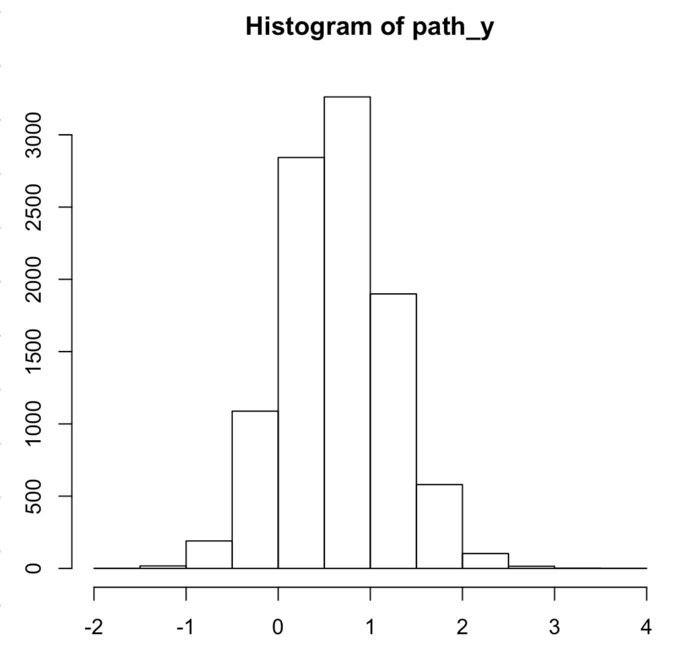
Generate $Y_{n+1} \sim N\left(\frac{1}{X_{n+1}}, \frac{1}{2(X_n+1)}\right)$

③ n++, goto ②.

```r
n = 10001
path_x = vector("numeric", n)
path_y = vector("numeric", n)
path_x[1] = 1
path_y[1] = 1

for (i in 2:n){
    x = path_x[i-1]
    y = path_y[i-1]
    x_new = rgamma(1, 3, 4+y^2)
    y_new = rnorm(1, 1/(x+1), sqrt(1/(2*(x+1))))
    path_x[i] = x_new
    path_y[i] = y_new
}
```

c). by b. hist shown that they converges. to gamma and Normal dist^{n}





d). by MC.

$E_\pi(X^2 Y^3 \exp(-X^2))$

$= \text{mean}(X^2 Y^3 \exp(-X^2))$ where $(X,Y) \sim \pi(x,y)$

$= 0.156$

Problem. 5

a) $E(Z) = 1.154$ $\quad E(1/Z) = 1.116025$.

set $\alpha = 2$, $\beta = 0.5$ for Gamma(α, β).

by independent MH algo.

① Let $n = 0$. $X_n = 1$

② generate $y \sim$ Gamma(α, β). $u \sim U(0,1)$

③ calculate $\alpha = \min\left\{\frac{f_z(y)\, g(x)}{f_z(x)\, g(y)}, 1\right\}$ and compare with u.

④ set X_{n+1}, $n{+}{+}$. go to ②.

```r
alpha = 2
beta = 0.5

theta1 = 1.5
theta2 = 2

fz = function(z){
    z^(-3/2)*exp(-theta1*z-theta2/z+2*sqrt(theta1*theta2)+log(sqrt(2*theta2)))
}

n = 1001
path = vector("numeric", n)
path[1] = 1

for (i in 2:n){
    y = rgamma(1, alpha, beta)
    u = runif(1)
    x = path[i-1]
    rate = min(fz(y)*dgamma(x, alpha, beta)/fz(x)/dgamma(y, alpha, beta), 1)
    if (u <= rate){
        path[i] = y
    } else {
        path[i] = x
    }
}
```

by R code: mean(path) = 1.098 $\quad$ mean(1/path) = 1.163

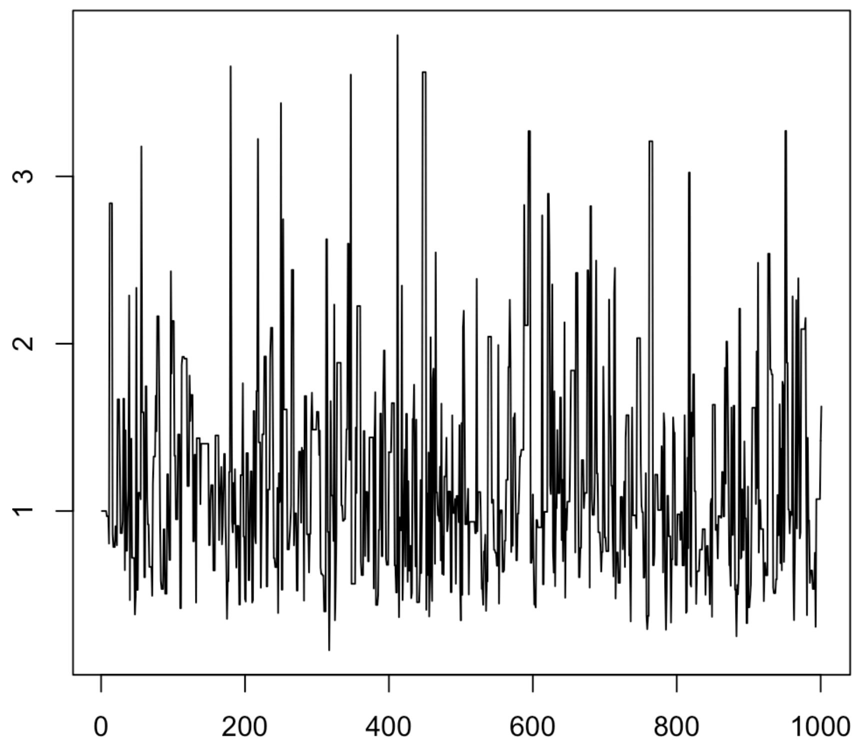
which are close to true value.

now try gamma(1.5, 2)

mean(path) = 1.196 mean(1/path) = 1.08

which is also close. But this gives higher accepting rate and from ts plot, the trace is more random.

(burn-in = 0)



b).

$$W = \log(Z)$$

$$F_w(W < w) = F(\log(z) < w) = F(z < e^w) = F_z(e^w)$$

$$\Rightarrow \frac{d}{dw} F_z(e^w) = e^w f_z(e^w)$$

$$f_w(w) \propto e^w (e^w)^{-\frac{3}{2}} \exp\left(-\theta_1 e^w - \frac{\theta_2}{e^w} + 2\sqrt{\theta_1\theta_2} + \log(\sqrt{2\theta_2})\right), \quad e^w > 0$$

by Random walk MH choose $g \overset{d}{\sim} N(0,1)$

① $n=0$, $X_n = 1$

② $\varepsilon_n \sim N(0,1)$, $u \sim \mathcal{U}(0,1)$

③ $y = X_n + \varepsilon_n$.

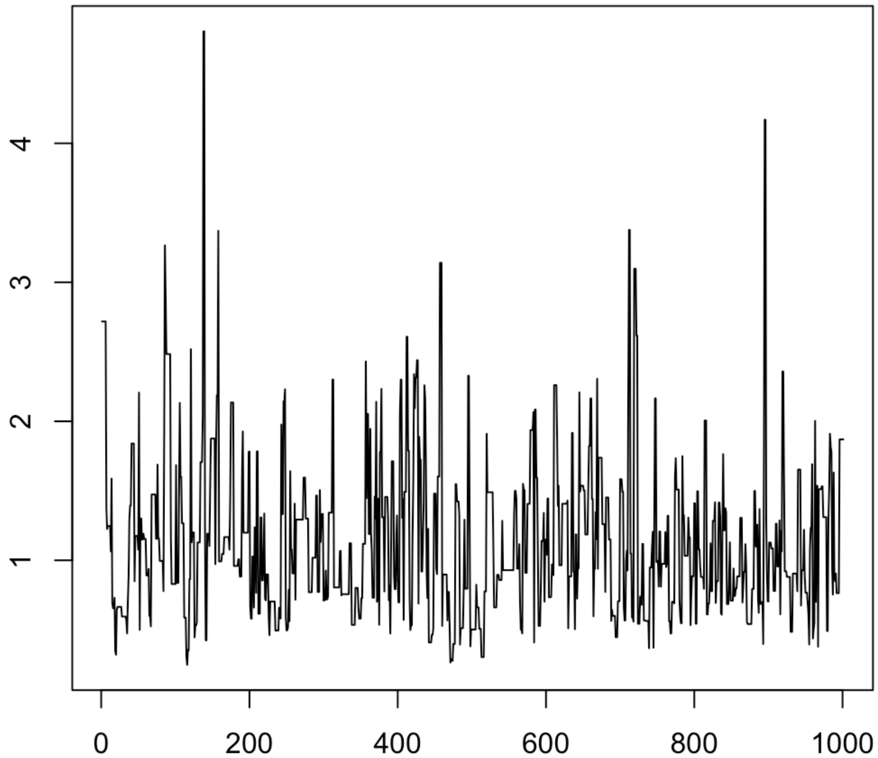
④ check if $u \leq \min\left\{\frac{f_w(y)}{f_w(X)}, 1\right\}$.

⑤ $n{+}{+}$, goto ①.

```r
fw = function(w){
    z = exp(w)
    return((z)*z^(-3/2)*exp(-theta1*z-theta2/z+2*sqrt(theta1*theta2)+log(sqrt(2*theta2))))
}

n = 1001
path = vector("numeric", n)
path[1] = 1

for (i in 2:n){
    x = path[i-1]
    eps = rnorm(1)
    u = runif(1)
    y = x + eps
    rate = min(fw(y)/fw(x), 1)
    if (u <= rate){
        path[i] = y
    } else {
        path[i] = x
    }
}

path = exp(path)
```

ts.plot:

(burn-in = 0)



mean(path) = 1.14494 mean(1/path) = 1.098067

which is very accurate compare to true value.

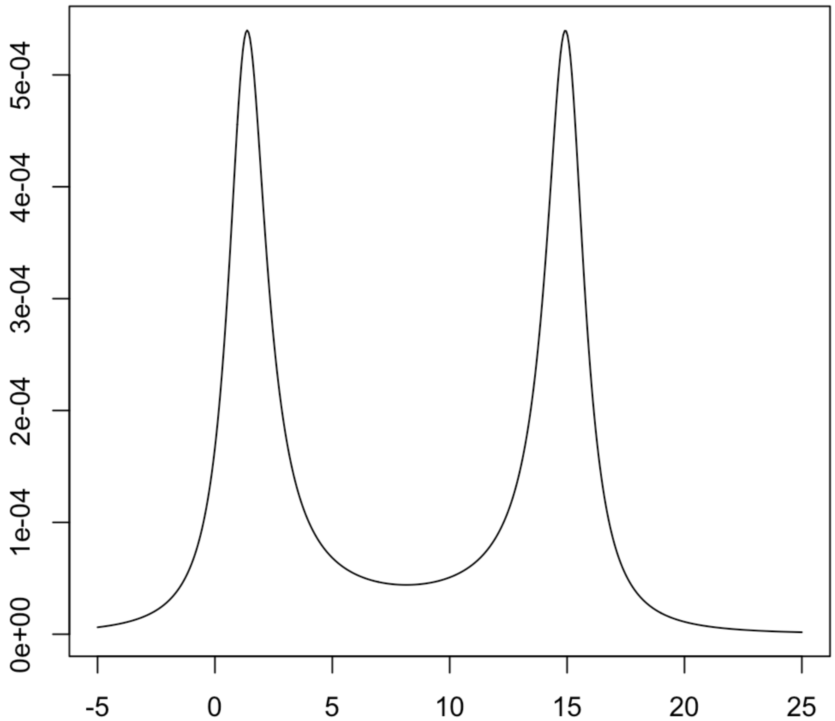
Problem 7.

cauchy $(\theta, 1)$ uniform $(0,1)$

a) posterior: $g(\theta | x_1, x_2) \propto f(x_1, x_2 | \theta)\, \pi(\theta)$

$$\propto \frac{1}{\pi^2 (1+(1.3-\theta)^2)(1+(15-\theta)^2)}$$

$\propto g(x)$



x.

b)

Metropolis Algo.

$\gamma = 1$

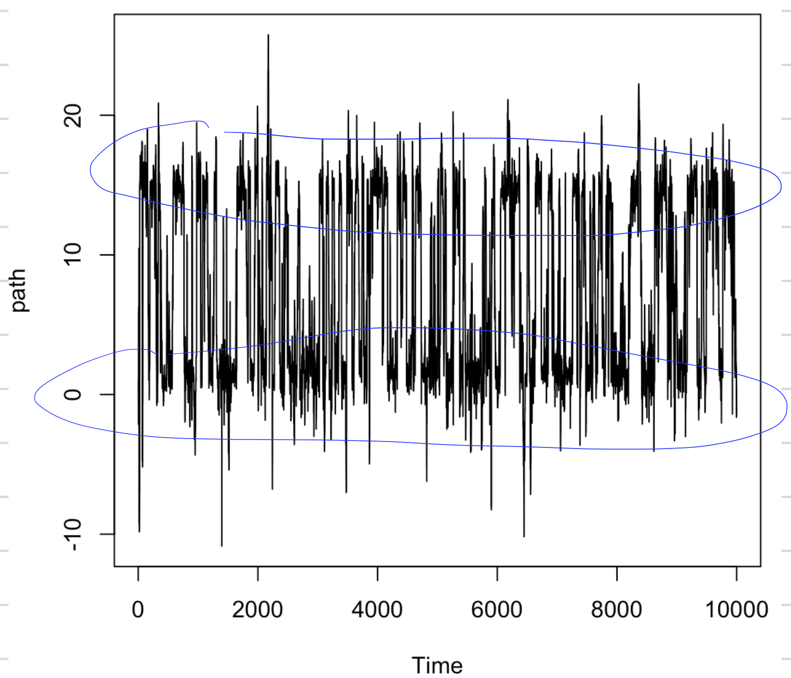
① $n=0$. $X_n = 1$

② gen. $y \sim \text{Cauchy}(X_n, \gamma)$, $u \sim U(0,1)$

③ if $r = \frac{\pi(y)}{\pi(X_n)} \geq u$. set $X_{n+1} = y$ else $X_{n+1} = X_n$.

④ n++ goto ②.

```r
scal = 1

n = 10001
path = vector("numeric", n)
path[1] = 1

for (i in 2:n){
    x = path[i-1]
    y = rcauchy(1, x, scal)
    u = runif(1)
    rate = min(g(y)/g(x), 1)
    if (u <= rate){
        path[i] = y
    } else {
        path[i] = x
    }
}
```



by tsplot we can see there is a high prob region at

$X = 15$ and $X = 1.5$.

c). tempering with ladder:

①. init vector ($x^{(0)}=1,\ i^{(0)}=1$), $n=0$.

②. gen $y \sim Cauchy(x^{(n)}, T_{i^{(n)}})$ ← which is $\pi_{i^{(n)}}(x^{(n)})$

by MH.

③. gen $i \sim q(i \mid i^{(n)}) = \begin{cases} 1 & \text{if } (i^{(n)}, i)=(1,2) \text{ or } (i^{(n)}, i)=(k,k-1) \\ \frac{1}{2} & \text{if } |i^{(n)}-i|=1 \text{ and } i^{(n)} \in \{2,\cdots,k-1\} \\ 0 & \text{else.} \end{cases}$

④. compute r, set $i^{(n+1)}$

⑤. go-to ②.

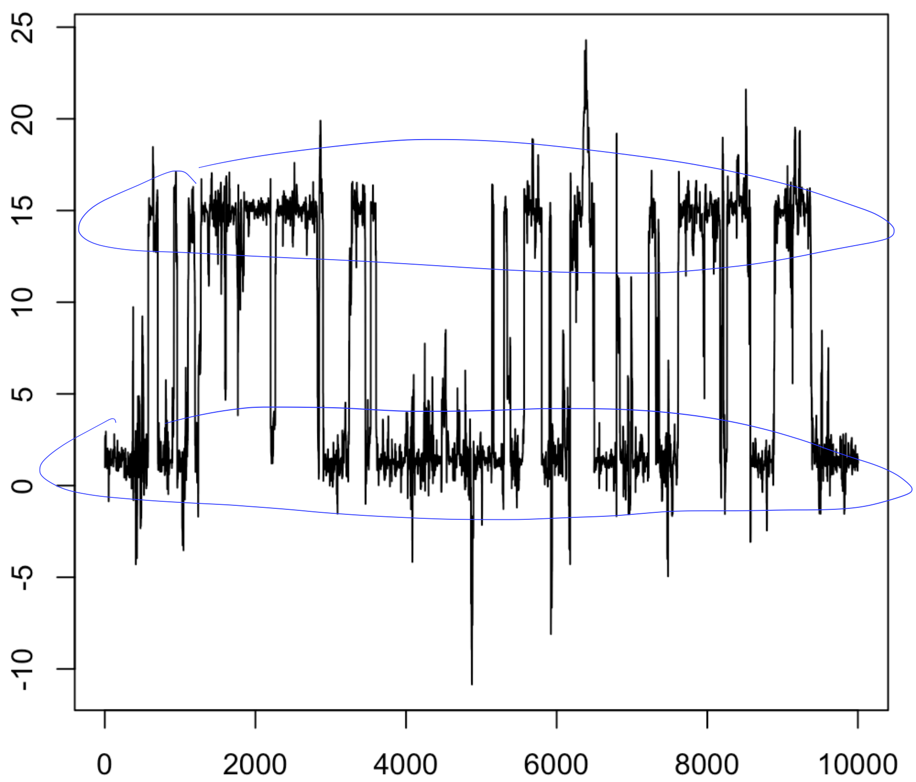
```r
n = 10001
path = vector("numeric", n)
i_vec = vector("numeric", n)
path[1] = 1
i_vec[1] = 1

T = seq(0.1, 1, 0.1)

for (i in 2:n) {
    x = path[i-1]
    y = rcauchy(1, x, T[i_vec[i-1]])
    u = runif(1)
    rate = g_new(y, T[i_vec[i-1]])/g_new(x, T[i_vec[i-1]])
    if (u <= rate){
        path[i] = y
    } else {
        path[i] = x
    }

    cur_i = 0
    case = 0

    if (i_vec[i-1] == 1) {cur_i = 2; case = 1}
    else if (i_vec[i-1] == 10) {cur_i = 9; case = 1}
    else {cur_i = i_vec[i-1] + sample(c(-1, 1), 1, TRUE, c(1/2, 1/2)); case = 2}

    top = dcauchy(path[i], x, T[cur_i]) * 1/10 * 1/case
    if(cur_i == 10 || cur_i == 10) {case = 1}
    bottom = dcauchy(path[i], x, T[i_vec[i-1]]) * 1/10 * 1/case

    r = top/bottom
    u = runif(1)
    if (r >= u) {i_vec[i] = cur_i}
    else {i_vec[i] = i_vec[i-1]}
}
```

↑ posterior dist$^{\underline{n}}$ of $Cauchy(\theta, T_{i^{(n-1)}})$

(define in next page.

```
g_new = function(theta, i) {
    1/(pi^2*(1+((1.3-theta)/i)^2)*(1+((15-theta)/i)^2))
}
```

ts.plot for tempering

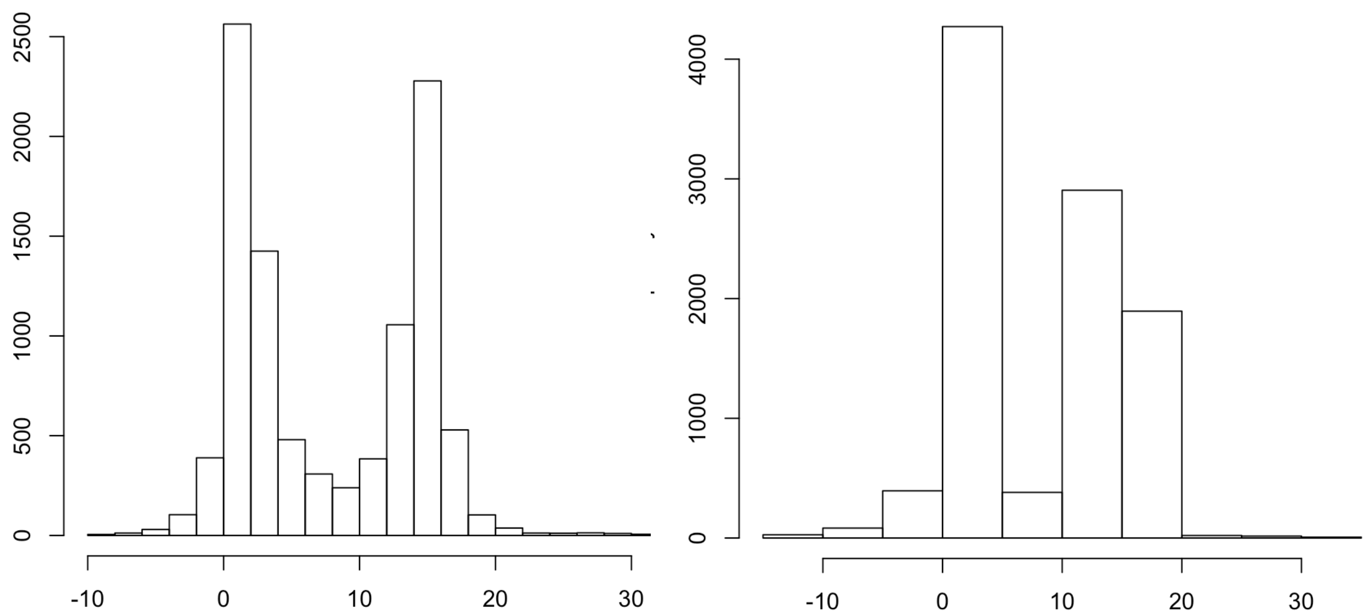


d). As we can see, both b) and c) generate

$X = 1.5, 15$ with high prob. which matches

the posterior dist^n we calculate in a). which has 2

mode.



They both has similar shape as a).

Problem 8.

a).

```r
y = c(40, 10, 21, 16, 11, 7, 4, 4, 2, 3)
x = c(30.11, 13.91, 10.63, 8.68, 7.14, 5.99, 5.23, 4.37, 4.09, 3.54)

model = glm(y~x, family = poisson(link = "identity"))

var = summary(model)$cov.unscaled
```

```
> var
            (Intercept)           x
(Intercept)   2.1875563 -0.25817166
x            -0.2581717  0.04363345
```

is the covariance matrix for $(\hat{\beta}_0, \hat{\beta}_1)$

b).

parametric: by a) we have $\hat{\beta}_0 = -2.312$

$\hat{\beta}_1 = 1.5063$

then we select x^* for x and gen $y^* = Poi(\hat{\beta}_0 + \hat{\beta}_1 x^*)$

```r
para_boostrap = function(x, y) {
    x_star = sample(x, 1000, TRUE)
    y_star = vector("numeric", 1000)
    for (i in 1:1000) {
        y_star[i] = rpois(1, -2.312+1.5063*x_star[i])
    }
    model = glm(y_star~x_star, family = poisson(link = "identity"))

    return(summary(model)$cov.unscaled)
}
```

```
> rv
            (Intercept)      x_star
(Intercept)  0.24610124 -0.02223001
x_star      -0.02223001  0.00333986
```

non-para.

select (x^*, y^*) and then estimate $\hat{\beta}_0, \hat{\beta}_1$

```
sample_idx = sample(c(1:10), 1000, TRUE)
x_star = x[sample_idx]
y_star = y[sample_idx]

model = glm(y_star~x_star, family = poisson(link = "identity"))

var = summary(model)$cov.unscaled
```

```
> var
            (Intercept)        x_star
(Intercept)  0.02277080 -0.0026917103
x_star      -0.00269171  0.0004508931
```

c). 95% CI.

non-para: Confint (model) by R = (1.466, 1.553)

para: Confint (model) by R = (1.490, 1.568)