Practice Questions

STAT 440/840 - CM 761

Questions

Q1

Suppose that $X_1, \ldots, X_n \mid \theta$ is an *iid* sample from the Poisson(θ) distribution. Suppose we obtain some data that is **summarized** in the following data.

```
x = rep( 0:5, times = c(13,  28, 29, 19, 8, 3) )
table( x )
```

```
## x
##  0  1  2  3  4  5
## 13 28 29 19  8  3
```

a) Show that the conjugate prior is a gamma distribution with hyperparameters (α, β).

b) Show that the posterior mean is a weighted average of the prior mean and maximum likelihood estimate.

c) Using each of the following priors caculate a credible interval for θ,

   i) a gamma prior with $\alpha = 1$ and $\beta = 10$,

   ii) a gamma prior with $\alpha = 10$ and $\beta = 1$,

   iii) a Jeffreys prior $p(\theta) \propto 1/\sqrt{\theta}$.

d) Using maximum likelihood calculate a confidence interval.

e) Compare and discuss the intervals generated in c) and d).

f) In c) we can calculate the credible intervals exactly. Instead use MCMC to estimate the credible interval using Jeffreys prior.

   i) Construct a `R` function that generates a random walk **MCMC** algorithm to sample from the posterior. The input is the random walk length, the standard deviation of the proposal density σ, and the initial position.

   ii) Run a random walk MCMC using $\sigma = 0.001, 0.01, 0.1$, $T = 10^4$ iterations and starting state $\theta^{(0)} = 1.9$. Then for each MCMC provide

   - traceplot, autocorrelation,
   - summary table with the acceptance rate, estimate of the posterior mean, naive estimate of the credible interval, and a measure of mixing using

$$\frac{1}{T}\sum_{t=1}^{T}(x_t - x_{t-1})^2$$

   - and then comment on the results.

Q2

Student's t-distribution with ν degrees of freedom has density function given by

$$f(x \mid \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Estimate the expectation and variance of a t-distribution with $\nu = 10$ degrees of freedom using a Metropolis-Hastings algorithm with the following **candidate densities**

a) $Y \sim N(0,1)$,

b) $Y = X_t + Z$ where $Z \sim N(0,1)$,

c) $Y \sim t_\nu$ with $\nu = 2$ and

d) $Y = X_t + E$ where $E \sim t_\nu$ with $\nu = 2$.

- Use 10,000 iterations. Summarize the results using
  - traceplots, autocorrelations and
  - the acceptance ratio.

Q3

MCMC Inference using Gibbs.

Consider the following posterior distribution:

$$\pi(\theta, \lambda, n|x, y, z) \propto \left(\begin{array}{c} n \\ \theta \end{array} \right) \lambda^{\theta+\alpha-1}(1-\lambda)^{n-\theta+\beta-1}\frac{\gamma^n}{n!}e^{-\gamma}.$$

(Note that here (α, β, γ) are essentially the data).

i) Give the conditional posterior distribution $\pi(\theta|\lambda, n, \alpha, \beta, \gamma)$.

ii) Give the conditional posterior distribution $\pi(\lambda|\theta, n, \alpha, \beta, \gamma)$.

iii) Give the conditional posterior distribution $\pi(n|\lambda, \theta, \alpha, \beta, \gamma)$.

(HINT: all the above can be found with standard forms.)

iv) Implement the above Gibbs sampler in `R` to sample from $\pi(\theta, \lambda, n|\alpha, \beta, \gamma)$ when $\alpha = \beta = \gamma = 1.5$.

Solutions

Q1

Suppose that $X_1, \ldots, X_n \mid \theta$ is an *iid* sample from the Poisson(θ) distribution. Suppose we have the following data.

```r
x = rep( 0:5, times = c(13,  28, 29, 19, 8, 3) )
table( x )
```

```
## x
##  0  1  2  3  4  5
## 13 28 29 19  8  3
```

a) Show that the conjugate prior is a gamma distribution with hyperparameters (α, β).

Let

$$\mathbf{x} = x_1, \ldots, x_n \qquad \text{and} \qquad \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

then the data conditional on the parameter θ

$$f(\mathbf{x} \mid \theta) = \prod_{i=1}^{n} \frac{\theta^{x_i} e^{-\theta}}{x_i!} = \theta^{n\overline{x}} e^{-n\theta} \times \prod_{i=1}^{n} \frac{1}{x_i!}$$

the proposed prior is a $\Gamma(\alpha, \beta)$

$$p(\theta \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta x}$$

the posterior is

$$\begin{aligned}
\pi(\theta|\mathbf{x}) &= f(\mathbf{x} \mid \theta) p(\theta \mid \alpha, \beta) / \pi(\mathbf{x} \mid \alpha, \beta) \\
&\propto f(\mathbf{x} \mid \theta) p(\theta \mid \alpha, \beta) \\
&\propto \theta^{n\overline{x}} e^{-n\theta} \times \prod_{i=1}^{n} \frac{1}{x_i!} \times \frac{\beta^\alpha}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta x} \\
&\propto \theta^{(\alpha+n\overline{x})-1} e^{-(n+\beta)\theta}
\end{aligned}$$

the posterior density is $\Gamma(\alpha + n\overline{x}, n + \beta)$.

b) Show that the posterior mean is a weighted average of the prior mean and maximum likelihood estimate.

- If $Y \sim \Gamma(\alpha, \beta)$ then $\mathbb{E}[Y] = \alpha/\beta$
- The maximum likelihood estimate is

$$\widehat{\theta} = \overline{x}$$

$$\theta|\mathbf{x} \sim \Gamma(\alpha + n\overline{x}, n + \beta)$$

$$\mathbb{E}[\theta \mid \mathbf{x}] = \frac{\alpha + n\overline{x}}{n + \beta} = \frac{\beta}{n+\beta}\frac{\alpha}{\beta} + \frac{n}{n+\beta}\overline{x} = \frac{\beta}{n+\beta}\frac{\alpha}{\beta} + \frac{n}{n+\beta}\widehat{\theta}$$

c) Using each of the following priors caculate a credible interval for θ,

i) a gamma prior with $\alpha = 1$ and $\beta = 10$,

ii) a gamma prior with $\alpha = 10$ and $\beta = 1$,

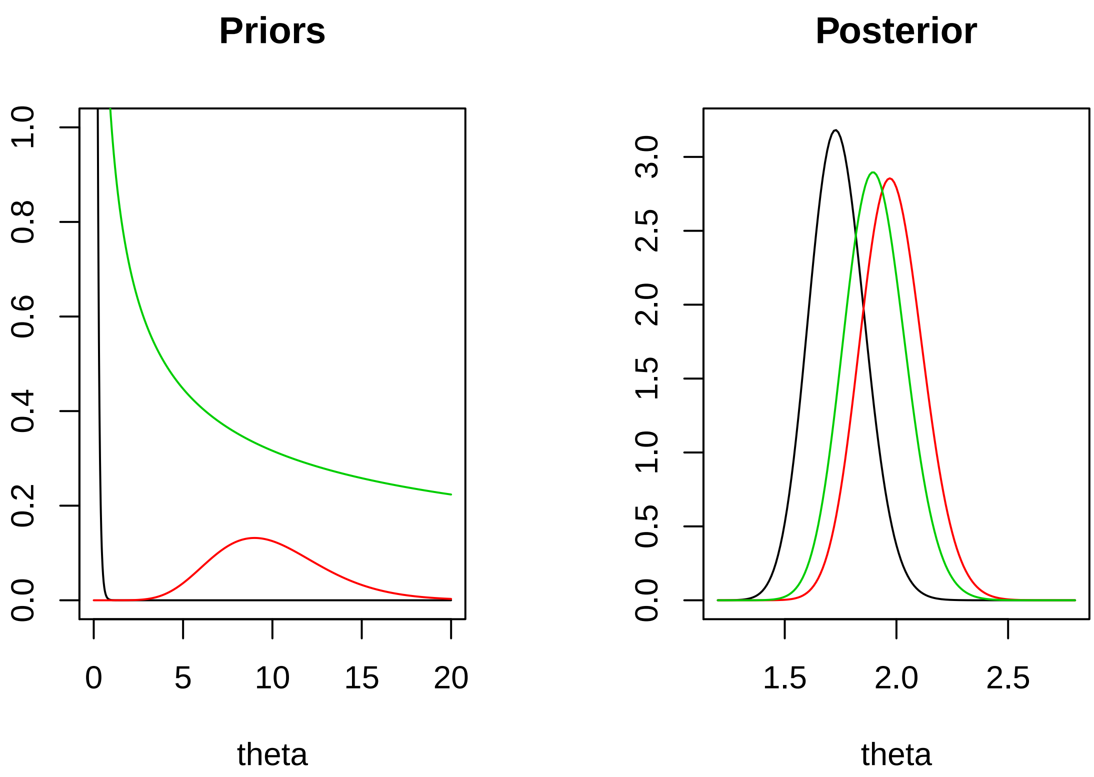
iii) a Jeffreys prior $p(\theta) \propto 1/\sqrt{\theta}$, The three intervals.

```
x = rep(0:5, c(13,28,29,19, 8, 3))

n = length(x)
theta.hat = mean(x)

#pab = matrix(0, nrow=3, ncol=2)
#pab[1,] = c(191, 110)
#pab[2,] = c(200, 101)
#pab[3,] = c(190.5, 100)
pab = matrix(0, nrow=3, ncol=2)
pab[1,] = c( 1, 10)
pab[2,] = c(10,  1)
pab[3,] = c(1/2, 0)

ab = pab + cbind( rep(theta.hat*n,3), rep(n,3) )

alpha = 0.05
aset= c( alpha/2, 1-alpha/2 )
ci.tab = rbind( qgamma( aset, shape=ab[1,1], rate=ab[1,2]),
   qgamma( aset, shape=ab[2,1], rate=ab[2,2]),
   qgamma( aset, shape=ab[3,1], rate=ab[3,2])  )

tab =  as.matrix(cbind(pab, ab, ab[,1]/ab[,2], ci.tab) )
dimnames(tab) = list( c("i)", "ii)", "iii)"), c("Prior-a", "Prior-b", "Post-a", "Post-b", "Post-Mean",
round(tab,2)
```

```
##      Prior-a Prior-b Post-a Post-b Post-Mean Lower Upper
## i)       1.0      10  191.0    110      1.74  1.50  1.99
## ii)     10.0       1  200.0    101      1.98  1.72  2.26
## iii)     0.5       0  190.5    100      1.90  1.64  2.18
```

a plot of the prior and the posterior.

```
par(mfrow=c(1,2))
z = seq(0, 20, 0.01)

plot(z, dgamma( z, shape=1, rate=10), type='l',ylim=c(0,1), main="Priors", xlab="theta", ylab="")
lines(z, dgamma( z, shape=10, rate=1 ), type='l', col=2)
lines(z, 1/sqrt(z), type='l', col=3)


z = seq(1.2, 2.8, 0.01)
plot(z, dgamma( z, shape=ab[1,1], rate=ab[1,2] ), type='l', main="Posterior", ylim=c(0,3.2), xlab="theta
lines(z, dgamma( z, shape=ab[2,1], rate=ab[2,2]), type='l', col=2)
lines(z, dgamma( z, shape=ab[3,1], rate=ab[3,2]), type='l', col=3)
```



d) Using maximum likelihood calculate a confidence interval.

They need to give the MLE and fisher for wald test (shown here) or use the liklihood ratio test.

CI based on the Wald test.

```
wald.ci = theta.hat + c(-1,1) * 1.96*sqrt(theta.hat/n)
wald.ci
```

```
## [1] 1.629833 2.170167
```

CI based on the log-likelihood ratio test.

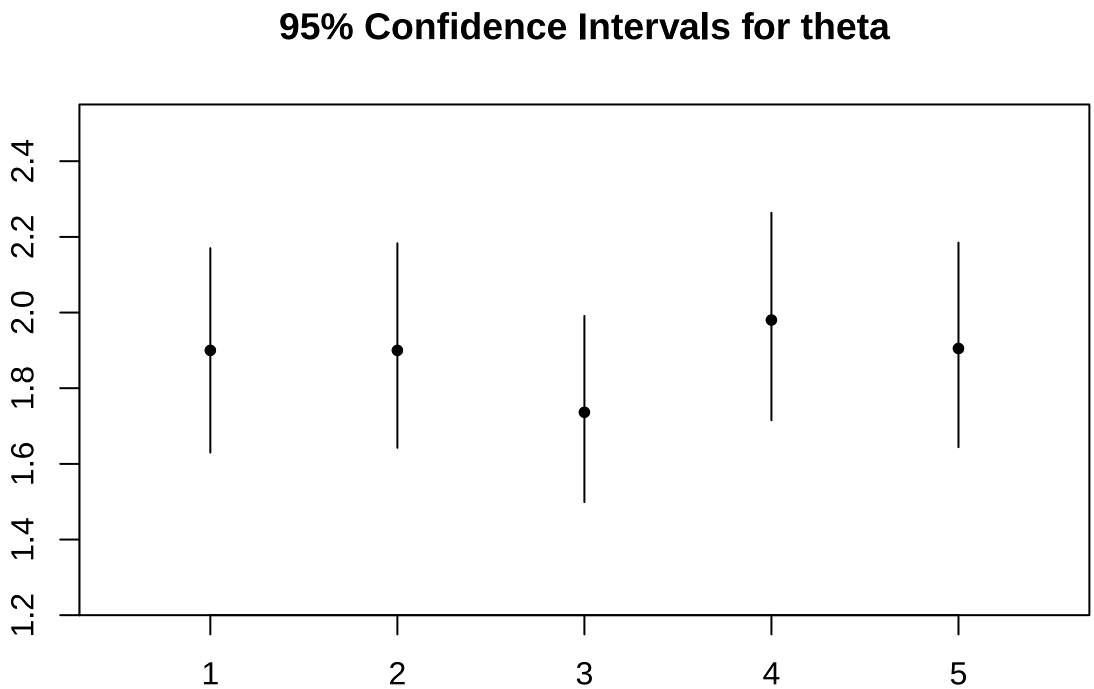
```
loglik <- function(lam=NULL, datax=NULL ) {
    val = numeric(length(lam))
    n = length(x)
    for (i in 1:length(lam)) {
        val[i] = - lam[i]*n + log(lam[i])*sum(x) - sum( lfactorial(x) )
    }
    return(val)
}

loglik.ratio <- function(x=NULL, datat=NULL, lam.mle=NULL, q0=NULL) {
  logratio = -2*(loglik(lam=x, datax=datat) - loglik(lam=lam.mle, datax=datat) )
    val = logratio - q0
    return(val)
}
```

```r
temp2 =uniroot( f= loglik.ratio, interval=c(theta.hat, 20), datat=datat, lam.mle=theta.hat, q0= 3.841459
temp1 =uniroot( f= loglik.ratio, interval=c(.1, theta.hat), datat=datat, lam.mle=theta.hat, q0= 3.841459
likratio.ci = c(temp1$root, temp2$root)
likratio.ci
```

```
## [1] 1.642518 2.183113
```

e) Compare and discuss the intervals generated in c) and d) .

```r
plot( x=c(1,1), y=wald.ci, ylim=c(1.25,2.5),xlim=c(1/2, 5.5), main="95% Confidence Intervals for theta"
lines(x=c(2,2), y=likratio.ci)

for (i in 1:3) lines(x=c(i,i)+2, y=ci.tab[i,])

points( 1:5, c(theta.hat, theta.hat, ab[,1]/ab[,2] )    , pch=20)
```




```r
tab = rbind(wald.ci, likratio.ci, ci.tab)
tab = as.matrix(cbind(c(theta.hat, theta.hat, ab[,1]/ab[,2] ), tab) )
dimnames(tab) = list( c("Wald", "likelihood Ratio", "Prior i)", "Prior ii)", "Prior iii)" ), c("Estimate
round(tab,2)
```

```
##                  Estimate Lower Upper
## Wald                 1.90  1.63  2.17
## likelihood Ratio     1.90  1.64  2.18
## Prior i)             1.74  1.50  1.99
## Prior ii)            1.98  1.72  2.26
## Prior iii)           1.90  1.64  2.18
```

The two intervals based on Jeffery's prior and the likelihood ratio (or Wald statistic) agree.

f) In c) we can calculate the credible intervals exactly. Instead use MCMC to estimate the credible interval using Jeffreys prior.

   i) Construct a `R` function that generates a random walk **MCMC** algorithm to sample from the posterior. The input is the random walk length, the standard deviation of the proposal density σ, and the initial position.

The posterior density is

$$\Gamma(\alpha + n\overline{x}, n + \beta)$$

Using Jeffreys prior we have that

$$\theta \mid x_1, \ldots, x_n \sim \Gamma(190.5, 100)$$

The target density

```
pi <- function(x) { dgamma( x, shape=190.5, rate=100) }
```

Random Walk Algorithm to Sample from Posterior

```
random.walk.mcmc <- function(n=NULL, sigma=1, x0) {
  x = numeric(n)
  x[1] = x0
  for (i in 2:n) {
    y = rnorm(1, x[i-1], sd=sigma)
    u = runif(1)
    accept = pi(y)/pi(x[i-1])
    if (u < accept) x[i] = y
    else x[i] = x[i-1]
  }
  return(x)
}
```

ii) Run a random walk MCMC using $\sigma = 0.001, 0.01, 0.1$, $T = 10^4$ iterations and starting state $\theta^{(0)} = 1.9$. Then for each MCMC provide

- traceplot, autocorrelation,
- summary table with the acceptance rate, estimate of the posterior mean, naive estimate of the credible interval, and a measure of mixing using

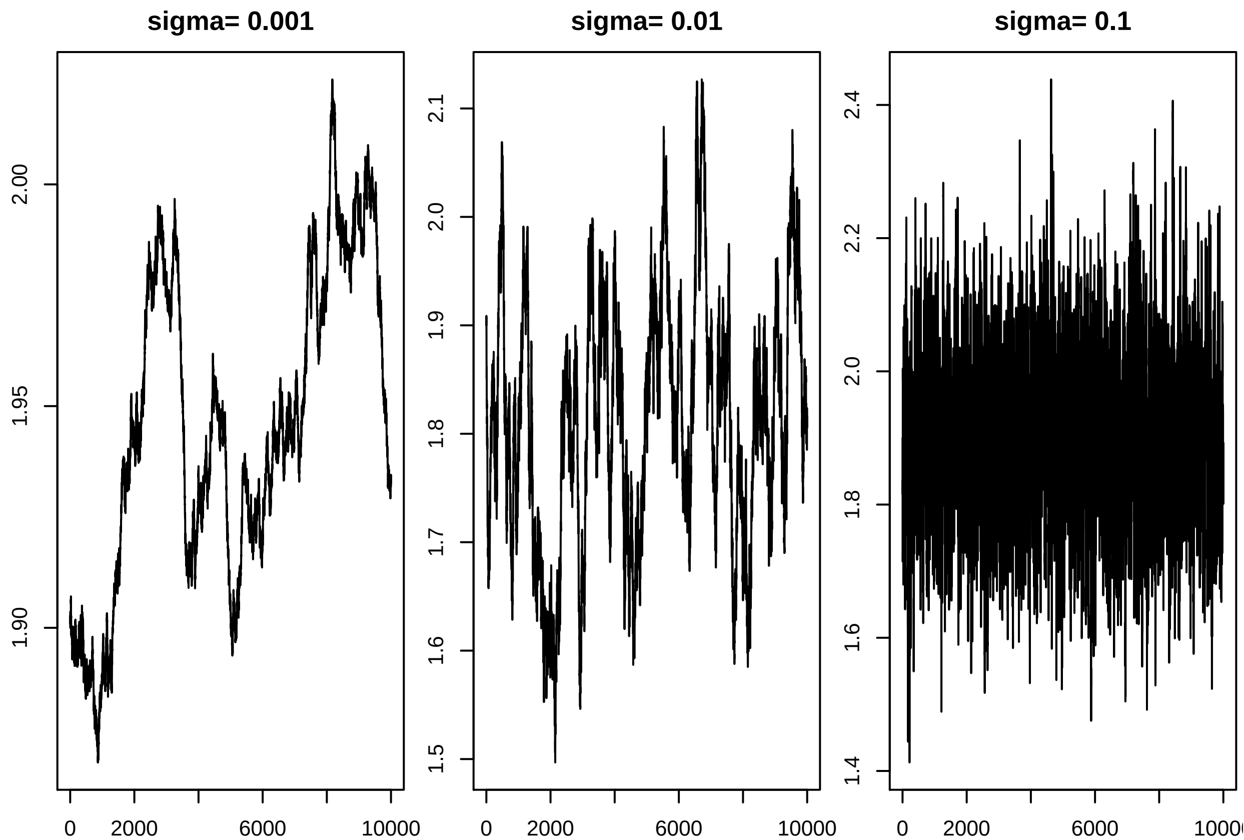
$$\frac{1}{T}\sum_{t=1}^{T}(x_t - x_{t-1})^2$$

- and then comment on the results.

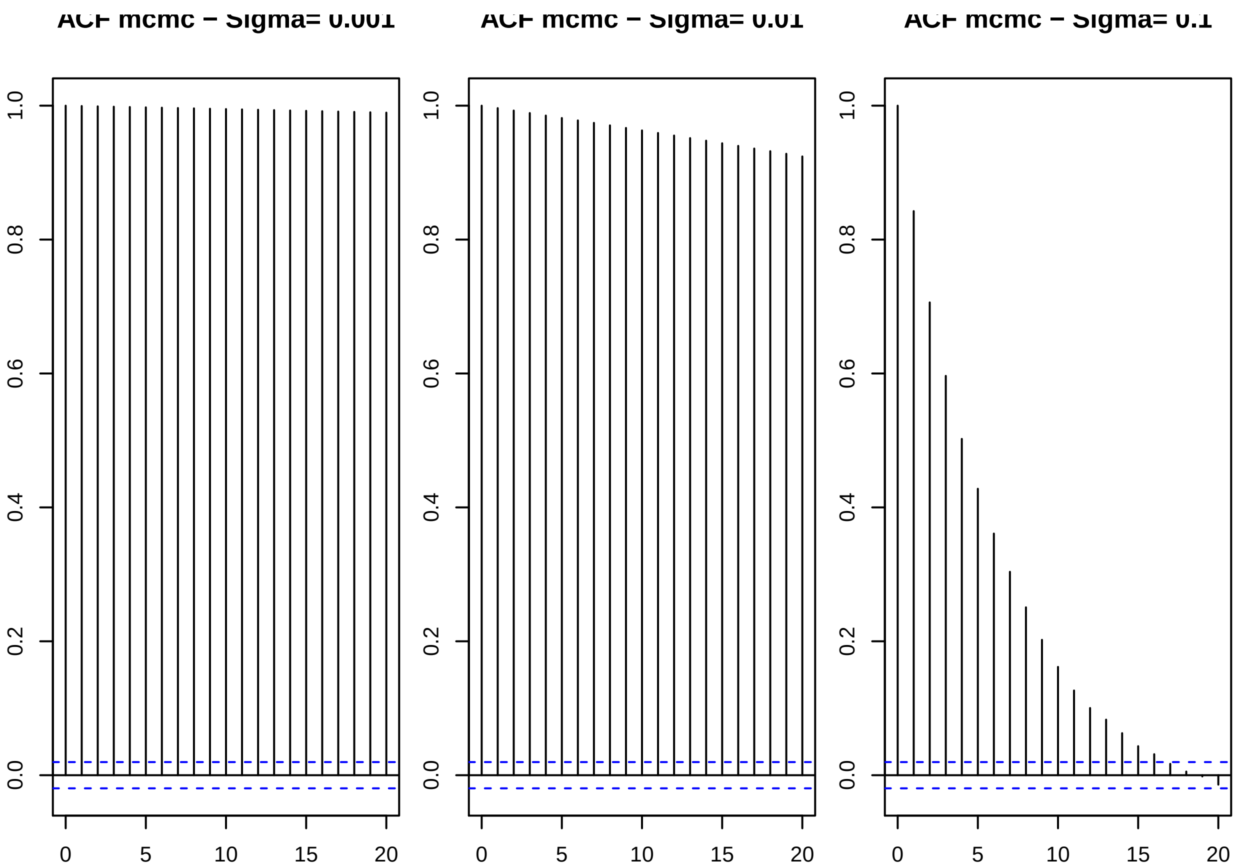
```
set.seed(440)
n0 = 10^4
x0 = 1.9
sigma = c(0.001, 0.01, 0.1)
mcmcx = matrix(0, nrow=n0, ncol=length(sigma))
for (i in 1:length(sigma)) mcmcx[,i]  = random.walk.mcmc(n=n0, sigma = sigma[i], x0=x0)
```

Trace and ACF Plots

```
par(mfrow=c(1,length(sigma)), mar=2.5*c(1,1,1,0.1))
for (i in 1:length(sigma)) plot(mcmcx[,i], main=paste('sigma=', sigma[i]), ylab="", type='l' )
```




```
for (i in 1:length(sigma)) acf( mcmcx[,i],  main=paste("ACF mcmc - Sigma=", sigma[i]), lag.max=20)
```




```
round( rbind(sigma, accept=apply(mcmcx, 2, function(z) { mean(z[-1] != z[-length(z)]) }), mixing=apply(
```

```
##           [,1]   [,2]   [,3]
## sigma  0.0010 0.0100 0.1000
## accept 0.9988 0.9728 0.7865
## mixing 0.0000 0.0001 0.0058
## 2.5%   1.8855 1.6027 1.6408
## 97.5%  2.0030 2.0405 2.1625
```

- From the traceplot and acf, we might need to increase σ in the random walk MCMC. OR it seems that $\sigma = 0.1$ is okay.
- The true confidence interval is (1.642.18). The MCMC random walk that used $\sigma = 0.1$ yields an interval close to this.

Q2

Student's t-distribution with ν degrees of freedom has density function given by

$$f(x \mid \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Estimate the expectation and variance of a t-distribution with $\nu = 10$ degrees of freedom using a Metropolis-Hastings algorithm with the following **candidate densities**

The target density is

```
pi <- function(x=NULL) { dt(x,df=10) }
```

a) $Y \sim N(0, 1)$,

```
mcmc1 <- function(n=NULL, x0=0) {
  x = numeric(n)
  x[1] = x0
  for (i in 2:n) {
    y = rnorm(1)
    u = runif(1)
    accept = pi(y)/pi(x[i-1]) * dnorm(x[i-1])/dnorm(y)
    if (u < accept) x[i] = y
    else x[i] = x[i-1]
  }
  return(x)
}
```

b) $Y = X_t + Z$ where $Z \sim N(0, 1)$,

```
mcmc2 <- function(n=NULL, x0=0) {
  x = numeric(n)
  x[1] = x0
  for (i in 2:n) {
    y = rnorm(1, x[i-1])
    u = runif(1)
    accept = pi(y)/pi(x[i-1])
    if (u < accept) x[i] = y
    else x[i] = x[i-1]
  }
  return(x)
}
```

c) $Y \sim t_\nu$ with $\nu = 2$ and

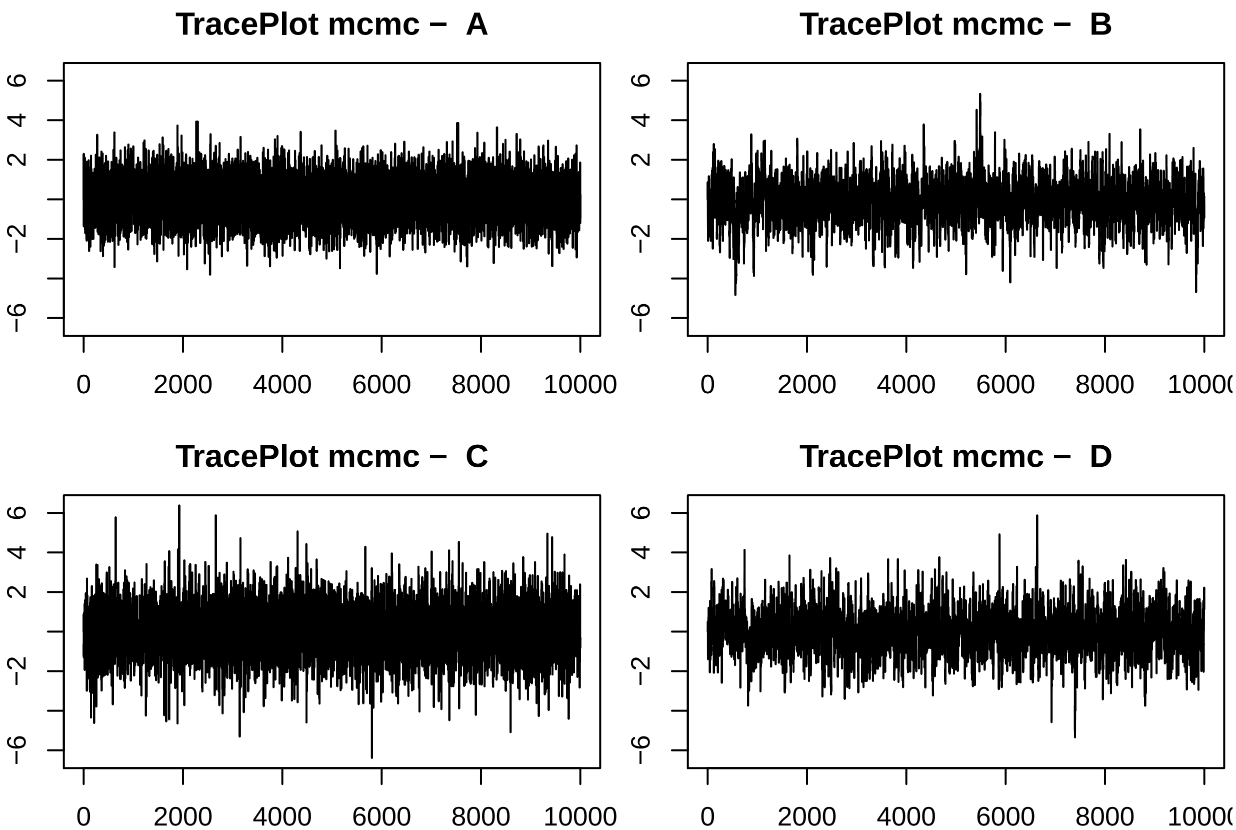
```
mcmc3 <- function(n=NULL, x0=0) {
  x = numeric(n)
  x[1] = x0
  for (i in 2:n) {
    y = rt(1, df=2)
    u = runif(1)
    accept = pi(y)/pi(x[i-1]) * dt(x[i-1], df=2)/dt(y,df=2)
    if (u < accept) x[i] = y
    else x[i] = x[i-1]
  }
  return(x)
}
```

d) $Y = X_t + Y$ where $Y \sim t_\nu$ with $\nu = 2$.

```
mcmc4 <- function(n=NULL, x0=0) {
  x = numeric(n)
  x[1] = x0
  for (i in 2:n) {
    y = x[i-1] + rt(1, df=2)
    u = runif(1)
    accept = pi(y)/pi(x[i-1])
    if (u < accept) x[i] = y
    else x[i] = x[i-1]
  }
  return(x)
}
```

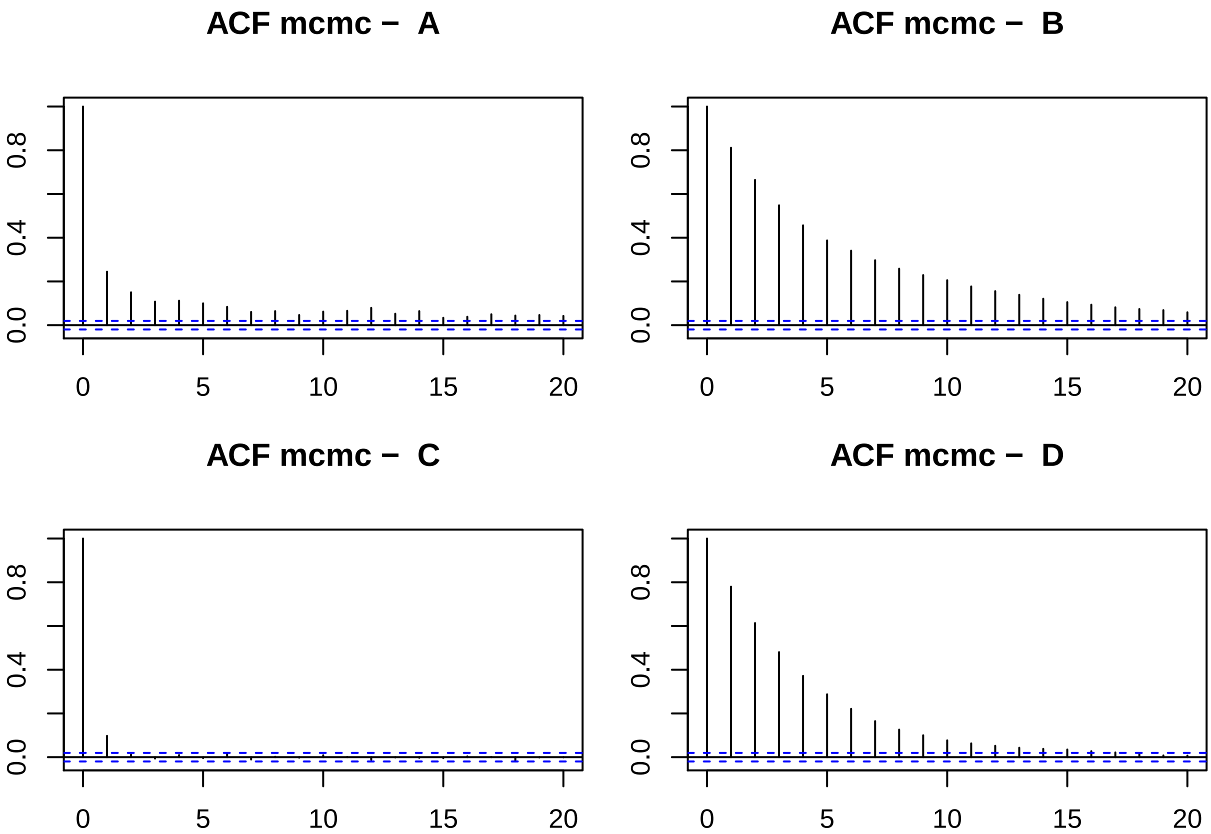
- Use 10,000 iterations. Summarize the results using
  - traceplots, autocorrelations and
  - the acceptance ratio.

```
n0 = 10^4
set.seed(440)
mcmc = matrix(0, nrow=n0, ncol=4)
mcmc[,1] = mcmc1(n0)
mcmc[,2] = mcmc2(n0)
mcmc[,3] = mcmc3(n0)
mcmc[,4] = mcmc4(n0)

par(mfrow=c(2,2), mar=2.5*c(1,1,1,0.1))
for (i in 1:4) plot( mcmc[,i], main=paste("TracePlot mcmc - ", LETTERS[i]), ylab="", type='l', ylim=rang
```



- From the trace plot for mcmc A, this MCMC does not expore the space as well as the others.
- MCMC C and D are mixing well.

```
par(mfrow=c(2,2), mar=3*c(1,1,1,0.1))
for (i in 1:4) acf( mcmc[,i],  main=paste("ACF mcmc - ", LETTERS[i]), lag.max=20)
```



- MCMC A and C have little auto-correlation

- MCMC B and D have some auto-correlation

```
round( rbind(1:4, accept.rate=apply(mcmc, 2, function(z) { mean(z[-1] != z[-length(z)]) }), average=appl
```

```
##              [,1]   [,2]   [,3]  [,4]
##             1.000  2.000  3.000 4.000
## accept.rate 0.958  0.711  0.869 0.627
## average     0.037 -0.115 -0.009 0.032
## variance    1.280  1.267  1.275 1.283
```

- MCMC A has a high acceptance however does not mix well.
- MCMC A underestimates the variance due to poor mixing.
- MCMC C, B, and D give resonable estimates.

The theoretical variance is

```
10/(10-2)
```

```
## [1] 1.25
```

Q3

MCMC Inference using Gibbs

Consider the following posterior distribution:

$$\pi(\theta, \lambda, n|x, y, z) \propto \left(\begin{array}{c} n \\ \theta \end{array} \right) \lambda^{\theta+\alpha-1}(1-\lambda)^{n-\theta+\beta-1}\frac{\gamma^n}{n!}e^{-\gamma}.$$

(Note that here (α, β, γ) are essentially the data).

i) Give the conditional posterior distribution $\pi(\theta|\lambda, n, \alpha, \beta, \gamma)$.

$$\theta|\left(\lambda, n, \alpha, \beta, \gamma\right) \sim \text{Bin}\left(n, \lambda\right)$$

ii) Give the conditional posterior distribution $\pi(\lambda|\theta, n, \alpha, \beta, \gamma)$.

$$\lambda|\left(\theta, n, \alpha, \beta, \gamma\right) \sim \text{BETA}\left(\theta+\alpha, n-\theta+\beta\right)$$

iii) Give the conditional posterior distribution $\pi(n|\lambda, \theta, \alpha, \beta, \gamma)$.

$$n|\left(\lambda, \theta, \alpha, \beta, \gamma\right) \sim \text{Poisson}\left(\gamma\left[1-\lambda\right]\right)+\theta$$

(HINT: all the above can be found with standard forms.)

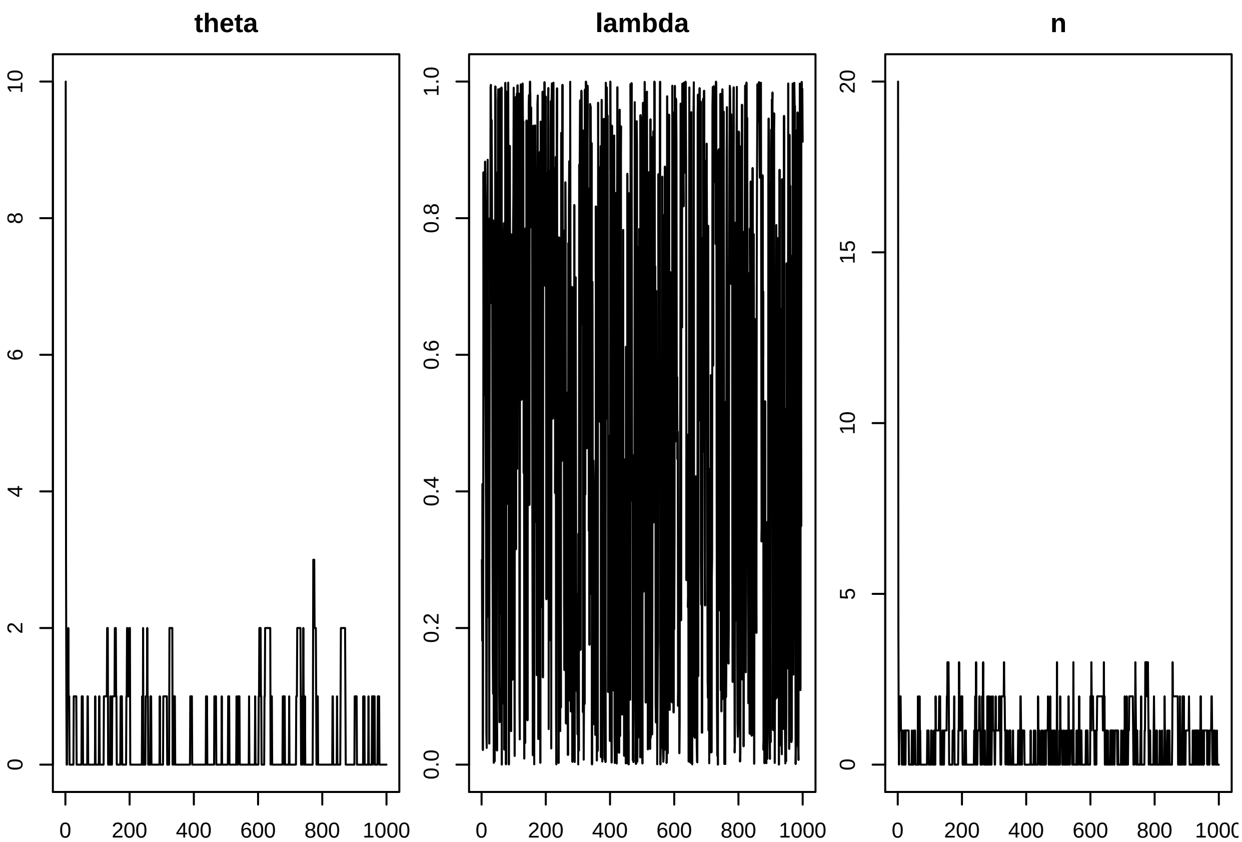
iv) Implement the above Gibbs sampler in R to sample from $\pi(\theta, \lambda, n|\alpha, \beta, \gamma)$ when $\alpha = \beta = \gamma = 1.5$.

$$\theta|\left(\lambda, n, \alpha, \beta, \gamma\right) \sim \text{Bin}\left(n, \lambda\right)$$
$$\lambda|\left(\theta, n, \alpha, \beta, \gamma\right) \sim \text{BETA}\left(\theta+1/2, n-\theta+1/2\right)$$
$$n|\left(\lambda, \theta, \alpha, \beta, \gamma\right) \sim \text{Poisson}\left(1/2\times\left[1-\lambda\right]\right)+\theta$$

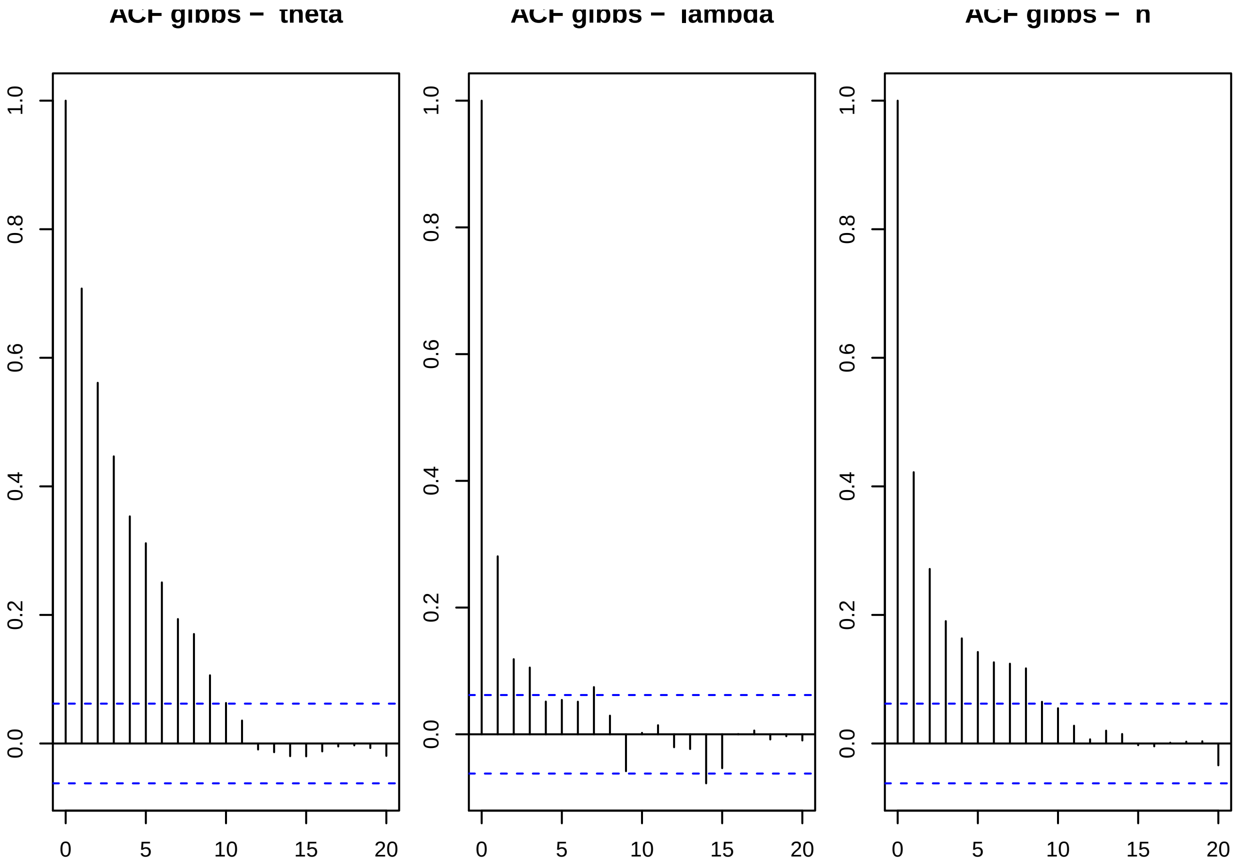
```
gibbs.mcmc <- function(n=NULL, par0=c(10,0.3,20) ){
  par = matrix(0, nrow=n, ncol=3)
  par = as.data.frame(par)
  names(par) = c("theta", "lambda", "n")
  par[1,] = par0
```

```
  for (i in 2:n){
    par$theta[i]  = rbinom(1, size=par$n[i-1], prob=par$lambda[i-1] )

    par$lambda[i] = rbeta(1, par$theta[i]+1/2, par$n[i-1]-par$theta[i]+1/2 )

    par$n[i]      = par$theta[i] + rpois(1, (1-par$lambda[i])/2 )
  }
  return(par)
  }
```

```
mcmcg = gibbs.mcmc(n = 1000)

par(mfrow=c(1,length(sigma)), mar=2.5*c(1,1,1,0.1))
for (i in 1:length(sigma)) plot(mcmcg[,i], main=paste(names(mcmcg)[i]), ylab="", type='l' )
```




```
for (i in 1:length(sigma)) acf( mcmcg[,i],  main=paste("ACF gibbs - ", names(mcmcg)[i]), lag.max=20)
```


ACF gibbs − theta
ACF gibbs − lambda
ACF gibbs − n
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20