

GRADUATE STUDENT STAT 840 A2

Vsevolod Ladtchenko 20895137

Problem 9

a)

Note that θ_i and ϕ_i are counts and therefore represent binomial probabilities.

$$\begin{aligned}\pi(\theta_i) &\propto \theta_i^{-1}(1 - \theta_i)^{-1} \\ \pi(\theta_i | X) &= f(X | \theta_i)\pi(\theta_i) \\ &= \binom{n}{k} \theta_i^k (1 - \theta_i)^{n-k} \pi(\theta_i) \\ &\propto \binom{n}{k} \theta_i^k (1 - \theta_i)^{n-k} \theta_i^{-1} (1 - \theta_i)^{-1} \\ &\propto \theta_i^{k-1} (1 - \theta_i)^{n-k-1} \\ &\propto \theta_i^{\alpha-1} (1 - \theta_i)^{\beta-1} \\ &\sim \text{Beta}(\alpha = k, \beta = n - k)\end{aligned}$$

For θ_1 , number of people with the disease (given D) who were exposed (E) is $k = 23$, and not exposed $n - k = 16$. Thus $\pi(\theta_1 | X) \sim \text{Beta}(23, 16)$.

For θ_2 , number of people with the disease (given D) and exposed (given E) who were apparently exposed (A) is $k = 18$, and apparently not exposed $n - k = 5$. Thus $\pi(\theta_2 | X) \sim \text{Beta}(18, 5)$.

For θ_3 , number of people with the disease (given D) and not exposed (given -E) who were apparently exposed (A) is $k = 3$, and apparently not exposed $n - k = 13$. Thus $\pi(\theta_3 | X) \sim \text{Beta}(3, 13)$.

For ϕ_1 , number of people without the disease (given -D) who were exposed (E) is $k = 32$, and not exposed $n - k = 44$. Thus $\pi(\phi_1 | X) \sim \text{Beta}(32, 44)$.

For ϕ_2 , number of people without the disease (given -D) and exposed (given E) who were apparently exposed (A) is $k = 16$, and apparently not exposed $n - k = 16$. Thus $\pi(\phi_2 | X) \sim \text{Beta}(16, 16)$.

For ϕ_3 , number of people without the disease (given -D) and not exposed (given -E) who were apparently exposed (A) is $k = 11$, and apparently not exposed $n - k = 33$. Thus $\pi(\phi_3 | X) \sim \text{Beta}(11, 33)$.

b)

First, a result:

$$\begin{aligned}
\frac{P(A \cap B \cap C)}{P(C)} &= \frac{P(A \cap B \cap C)}{P(C)} \\
P(A \cap B \mid C) &= \frac{P(A \cap B \cap C)}{P(C)} \frac{P(B \cap C)}{P(B \cap C)} \\
&= \frac{P(A \cap B \cap C)}{P(B \cap C)} \frac{P(B \cap C)}{P(C)} \\
&= P(A \mid B \cap C) P(B \mid C) \\
P(A, B \mid C) &= P(A \mid B, C) P(B \mid C)
\end{aligned}$$

Now the question:

$$\begin{aligned}
P(A, D) &= P(A, D, E) + P(A, D, \bar{E}) \\
&\quad \text{law of total probability} \\
P(A \mid D) P(D) &= P(A, E \mid D) P(D) + P(A, \bar{E} \mid D) P(D) \\
&\quad \text{definition of conditional probability} \\
P(A \mid D) &= P(A, E \mid D) + P(A, \bar{E} \mid D) \\
&\quad \text{assume } P(D) > 0 \\
&= P(A \mid E, D) P(E \mid D) + P(A \mid \bar{E}, D) P(\bar{E} \mid D) \\
&\quad \text{above result} \\
&= \theta_2 \theta_1 + \theta_3 (1 - P(E \mid D)) \\
&= \theta_2 \theta_1 + \theta_3 (1 - \theta_1) \\
&\quad \text{as defined in the question} \\
P(D) &= P(A, D) + P(\bar{A}, D) \\
&\quad \text{law of total probability} \\
&= P(A \mid D) P(D) + P(\bar{A} \mid D) P(D) \\
&\quad \text{definition of conditional probability} \\
1 &= P(A \mid D) + P(\bar{A} \mid D) \\
&\quad \text{assume } P(D) > 0 \\
P(\bar{A} \mid D) &= 1 - P(A \mid D) \\
&= 1 - \theta_2 \theta_1 - \theta_3 (1 - \theta_1)
\end{aligned}$$

Similarly, we can get:

$$\begin{aligned}
P(A \mid \bar{D}) &= P(A \mid E, \bar{D}) P(E \mid \bar{D}) + P(A \mid \bar{E}, \bar{D}) P(\bar{E} \mid \bar{D}) \\
&= \phi_2 \phi_1 + \phi_3 (1 - \phi_1) \\
P(\bar{A} \mid \bar{D}) &= 1 - \phi_2 \phi_1 - \phi_3 (1 - \phi_1)
\end{aligned}$$

Let X_1 be the first dataset, and X_2 be the second dataset.

$$\begin{aligned}
\pi(\theta_i | X_1, X_2) &= f(X_2 | \theta_i) \pi(\theta_i | X_1) \\
&\quad \text{note conditional independence: } f(X_2 | \theta_i) = f(X_2 | X_1, \theta_i) \\
f(X_2 | \theta_i) &= \binom{n}{k} P(A | D)^k (1 - P(A | D))^{n-k} \\
&= \binom{n}{k} P(A | D)^k P(\bar{A} | D)^{n-k} \\
\pi(\theta_i | X_1, X_2) &\propto \binom{n}{k} P(A | D)^k P(\bar{A} | D)^{n-k} \text{Beta}(\alpha_{\theta_i}, \beta_{\theta_i}) \\
&\propto \left(\theta_2 \theta_1 + \theta_3 (1 - \theta_1) \right)^{375} \left(1 - \theta_2 \theta_1 - \theta_3 (1 - \theta_1) \right)^{318} \theta_i^{\alpha_{\theta_i} - 1} (1 - \theta_i)^{\beta_{\theta_i} - 1} \\
\pi(\phi_i | X_1, X_2) &\propto \binom{n}{k} P(A | \bar{D})^k P(\bar{A} | \bar{D})^{n-k} \text{Beta}(\alpha_{\phi_i}, \beta_{\phi_i}) \\
&\propto \left(\phi_2 \phi_1 + \phi_3 (1 - \phi_1) \right)^{535} \left(1 - \phi_2 \phi_1 - \phi_3 (1 - \phi_1) \right)^{701} \phi_i^{\alpha_{\phi_i} - 1} (1 - \phi_i)^{\beta_{\phi_i} - 1}
\end{aligned}$$

Now consider evaluating the acceptance probability using these posteriors. Due to the high powers, we probably will not get accurate floating point results. Consider taking the log, which is an increasing monotone function. Since $\log(0)$ in R is $-\text{Inf}$, it will still work in comparisons.

$$\begin{aligned}
U &\leq \alpha(\theta_n, \theta^*) \\
&\leq \min \left\{ \frac{\pi(\theta^* | X) q(\theta^*, \theta_n)}{\pi(\theta_n | X) q(\theta_n, \theta^*)}, 1 \right\} \\
\log U &\leq \log \alpha(\theta_n, \theta^*) \\
&\leq \log \min \left\{ \frac{\pi(\theta^* | X) q(\theta^*, \theta_n)}{\pi(\theta_n | X) q(\theta_n, \theta^*)}, 1 \right\} \\
&\leq \min \left\{ \log \frac{\pi(\theta^* | X) q(\theta^*, \theta_n)}{\pi(\theta_n | X) q(\theta_n, \theta^*)}, 0 \right\}
\end{aligned}$$

The acceptance ratio becomes:

$$\begin{aligned}
\log \frac{\pi(\theta^* | X) q(\theta^*, \theta_n)}{\pi(\theta_n | X) q(\theta_n, \theta^*)} &= \log \pi(\theta^* | X) q(\theta^*, \theta_n) - \log \pi(\theta_n | X) q(\theta_n, \theta^*) \\
&= \log \pi(\theta^* | X) + \log q(\theta^*, \theta_n) - \log \pi(\theta_n | X) - \log q(\theta_n, \theta^*)
\end{aligned}$$

The posteriors become:

$$\begin{aligned}\log \pi(\theta_i | X) &= 375 \log(\theta_2 \theta_1 + \theta_3(1 - \theta_1)) + 318 \log(1 - \theta_2 \theta_1 - \theta_3(1 - \theta_1)) + (\alpha_{\theta_i} - 1) \log \theta_i + (\beta_{\theta_i} - 1) \log(1 - \theta_i) \\ \log \pi(\phi_i | X) &= 535 \log(\phi_2 \phi_1 + \phi_3(1 - \phi_1)) + 701 \log(1 - \phi_2 \phi_1 - \phi_3(1 - \phi_1)) + (\alpha_{\phi_i} - 1) \log \phi_i + (\beta_{\phi_i} - 1) \log(1 - \phi_i) \\ \log \pi(\theta_1 | X) &= 375 \log(\theta_2 \theta_1 + \theta_3(1 - \theta_1)) + 318 \log(1 - \theta_2 \theta_1 - \theta_3(1 - \theta_1)) + 22 \log \theta_1 + 15 \log(1 - \theta_1) \\ \log \pi(\theta_2 | X) &= 375 \log(\theta_2 \theta_1 + \theta_3(1 - \theta_1)) + 318 \log(1 - \theta_2 \theta_1 - \theta_3(1 - \theta_1)) + 17 \log \theta_2 + 4 \log(1 - \theta_2) \\ \log \pi(\theta_3 | X) &= 375 \log(\theta_2 \theta_1 + \theta_3(1 - \theta_1)) + 318 \log(1 - \theta_2 \theta_1 - \theta_3(1 - \theta_1)) + 2 \log \theta_3 + 12 \log(1 - \theta_3) \\ \log \pi(\phi_1 | X) &= 535 \log(\phi_2 \phi_1 + \phi_3(1 - \phi_1)) + 701 \log(1 - \phi_2 \phi_1 - \phi_3(1 - \phi_1)) + 31 \log \phi_1 + 43 \log(1 - \phi_1) \\ \log \pi(\phi_2 | X) &= 535 \log(\phi_2 \phi_1 + \phi_3(1 - \phi_1)) + 701 \log(1 - \phi_2 \phi_1 - \phi_3(1 - \phi_1)) + 15 \log \phi_2 + 15 \log(1 - \phi_2) \\ \log \pi(\phi_3 | X) &= 535 \log(\phi_2 \phi_1 + \phi_3(1 - \phi_1)) + 701 \log(1 - \phi_2 \phi_1 - \phi_3(1 - \phi_1)) + 10 \log \phi_3 + 32 \log(1 - \phi_3)\end{aligned}$$

```

log_post_theta_1 = function(p1,p2,p3)
{
  part1 = 375*log(p2*p1 + p3*(1-p1))
  part2 = 318*log(1 - p2*p1 - p3*(1-p1))
  part3 = 22*log(p1) + 15*log(1-p1)
  return(part1 + part2 + part3)
}

log_post_theta_2 = function(p1,p2,p3)
{
  part1 = 375*log(p2*p1 + p3*(1-p1))
  part2 = 318*log(1 - p2*p1 - p3*(1-p1))
  part3 = 17*log(p2) + 4*log(1-p2)
  return(part1 + part2 + part3)
}

log_post_theta_3 = function(p1,p2,p3)
{
  part1 = 375*log(p2*p1 + p3*(1-p1))
  part2 = 318*log(1 - p2*p1 - p3*(1-p1))
  part3 = 2*log(p3) + 12*log(1-p3)
  return(part1 + part2 + part3)
}

log_post_phi_1 = function(p1,p2,p3)
{
  part1 = 535*log(p2*p1 + p3*(1-p1))
  part2 = 701*log(1 - p2*p1 - p3*(1-p1))
  part3 = 31*log(p1) + 43*log(1-p1)
  return(part1 + part2 + part3)
}

log_post_phi_2 = function(p1,p2,p3)

```

```

{
    part1 = 535*log(p2*p1 + p3*(1-p1))
    part2 = 701*log(1 - p2*p1 - p3*(1-p1))
    part3 = 15*log(p2) + 15*log(1-p2)
    return(part1 + part2 + part3)
}

log_post_phi_3 = function(p1,p2,p3)
{
    part1 = 535*log(p2*p1 + p3*(1-p1))
    part2 = 701*log(1 - p2*p1 - p3*(1-p1))
    part3 = 10*log(p3) + 32*log(1-p3)
    return(part1 + part2 + part3)
}

#####
#####

log_accept_theta_1 = function(x, y, theta2, theta3)
{
    # log q = 0 since q = 1 for uniform density
    r1 = log_post_theta_1(y, theta2, theta3) - log_post_theta_1(x, theta2, theta3)
    return(min(r1, 0))
}

log_accept_theta_2 = function(x, y, theta1, theta3)
{
    r1 = log_post_theta_2(theta1, y, theta3) - log_post_theta_2(theta1, x, theta3)
    return(min(r1, 0))
}

log_accept_theta_3 = function(x, y, theta1, theta2)
{
    r1 = log_post_theta_3(theta1, theta2, y) - log_post_theta_3(theta1, theta2, x)
    return(min(r1, 0))
}

log_accept_phi_1 = function(x, y, phi2, phi3)
{
    r1 = log_post_phi_1(y, phi2, phi3) - log_post_phi_1(x, phi2, phi3)
    return(min(r1, 0))
}

log_accept_phi_2 = function(x, y, phi1, phi3)
{
    r1 = log_post_phi_2(phi1, y, phi3) - log_post_phi_2(phi1, x, phi3)
    return(min(r1, 0))
}

log_accept_phi_3 = function(x, y, phi1, phi2)
{
    r1 = log_post_phi_3(phi1, phi2, y) - log_post_phi_3(phi1, phi2, x)
    return(min(r1, 0))
}

```

```
}
```

We will now perform a series of experiments. Because it is hard to imagine a good joint density as the proposal density, we will skip regular Metropolis-Hastings.

The next option is Independent MH. Because the posterior density includes a Beta as a component, we will consider distributions on the support $[0, 1]$. One option is to choose the Uniform distribution, in order to be as “uninformative” as possible in regards to the final distribution’s shape. It will also have the effect of canceling out in the acceptance ratio calculation. Another option is to use the prior Beta distribution of each parameter. This will require modifying the acceptance functions.

We will also try the Random Walk MH, which turns out to have the same acceptance function as the Independent MH with Uniform proposal. We just need to make sure that Y does not exceed 0, 1.

Because of the way we derived the posterior densities, it seems more natural to perform the MH steps component-wise. We cannot optimize the parameters individually because they all depend on each other.

The Gibbs Sampler is not an option because it requires sampling from the posterior, which we cannot do. The other methods simply require evaluating it.

```
attempt_3_random_walk = function(n = 10000)
{
  chain = matrix(ncol = 6, nrow = n)
  chain[1,] = 0.5

  for (i in 2:n)
  {
    # the parameters, get updated with chain within the iteration
    theta1 = chain[i-1,1]
    theta2 = chain[i-1,2]
    theta3 = chain[i-1,3]
    phi1 = chain[i-1,4]
    phi2 = chain[i-1,5]
    phi3 = chain[i-1,6]

    x = chain[i-1,]
    e = rnorm(6, mean=0, sd=0.05)
    y = pmax(pmin(x + e, 1), 0) # candidate
    logu = log(runif(6)) # compare to log acceptance ratio for stability

    # theta_1
    if (logu[1] <= log_accept_theta_1(x[1], y[1], theta2, theta3))
    {
      chain[i,1] = y[1]
      theta1 = y[1]
    }
    else
      chain[i,1] = x[1]

    # theta_2
    if (logu[2] <= log_accept_theta_2(x[2], y[2], theta1, theta3))
    {
      chain[i,2] = y[2]
      theta2 = y[2]
    }
    else
```

```

    chain[i,2] = x[2]

    # theta_3
    if (logu[3] <= log_accept_theta_3(x[3], y[3], theta1, theta2))
    {
        chain[i,3] = y[3]
        theta3 = y[3]
    }
    else
        chain[i,3] = x[3]

    # phi_1
    if (logu[4] <= log_accept_phi_1(x[4], y[4], phi2, phi3))
    {
        chain[i,4] = y[4]
        phi1 = y[4]
    }
    else
        chain[i,4] = x[4]

    # phi_2
    if (logu[5] <= log_accept_phi_2(x[5], y[5], phi1, phi3))
    {
        chain[i,5] = y[5]
        phi2 = y[5]
    }
    else
        chain[i,5] = x[5]

    # phi_3
    if (logu[6] <= log_accept_phi_3(x[6], y[6], phi1, phi2))
    {
        chain[i,6] = y[6]
        phi3 = y[6]
    }
    else
        chain[i,6] = x[6]
}
return(chain)
}

```

For completeness, here are the lines of code to change to use the Beta prior as proposal:

```

log_accept_phi_3 = function(x, y, phi1, phi2) # also edit the other 5 functions
{
    r1 = log_post_phi_3(phi1, phi2, y) - log_post_phi_3(phi1, phi2, x)
    r1 = r1 + log(dbeta(x, 11,33)) - log(dbeta(y, 11,33))
    return(min(r1, 0))
}

y1 = rbeta(1, 23,16)
y2 = rbeta(1, 18,5)
y3 = rbeta(1, 3,13)
y4 = rbeta(1, 32,44)

```

```

y5 = rbeta(1, 16,16)
y6 = rbeta(1, 11,33)
y = c(y1,y2,y3,y4,y5,y6) # candidate

```

And here are the lines to change to use the Uniform proposal:

```

y = runif(6) # candidate

```

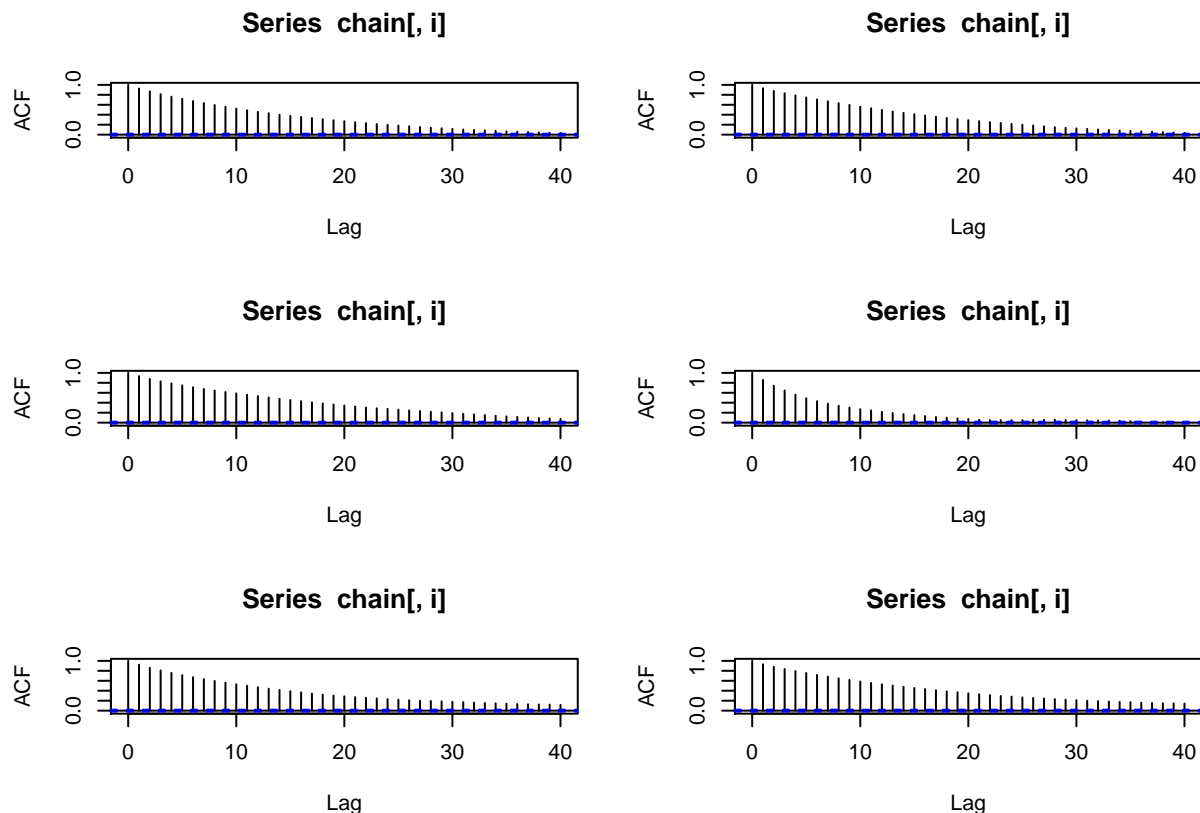
For the uniform proposal, the histograms look smooth and are consistent between runs. The plots of the chain also look fine but have a lot of flat regions. This means a lot of times we fail to accept a new point. This may be due to the choice of proposal distribution: choosing anything but uniform would imply prior knowledge of the shape of the distribution, as Beta distributions vary wildly in shape. Wanting to remain neutral in this regard, of course the uniform distribution will provide a lot of “misses” when we evaluate the height of the density at the next step, since it jumps around uniformly. Due to these long flat regions, we get very high autocorrelation in the ACF plot.

The marginal densities, time series of the chain, and ACF plots were similar for all experiments, but the ACF is slightly better for the random walk MH, so that is the one we show.

```

chain = attempt_3_random_walk(n = 10000)
par(mfrow=c(3,2))
for (i in 1:6) acf(chain[,i])

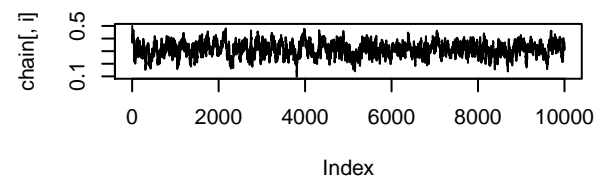
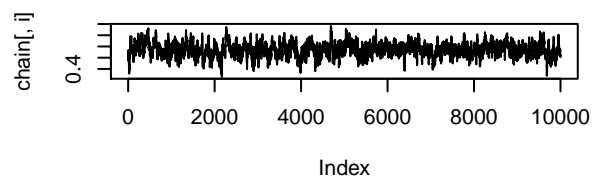
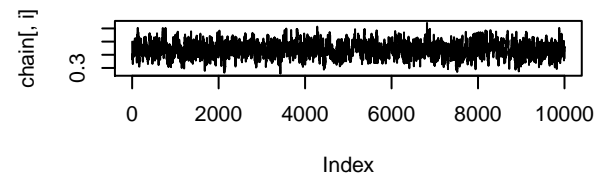
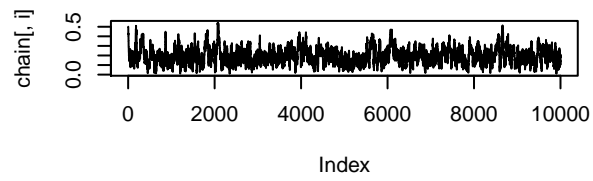
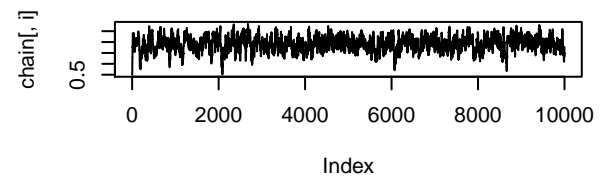
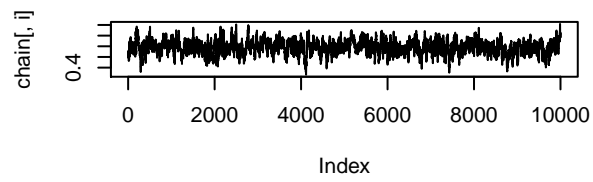
```



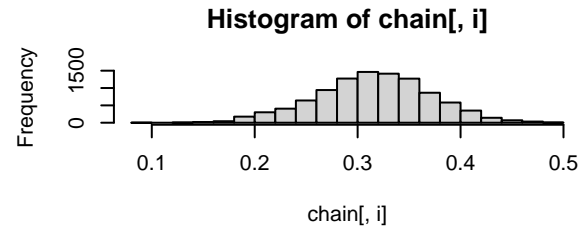
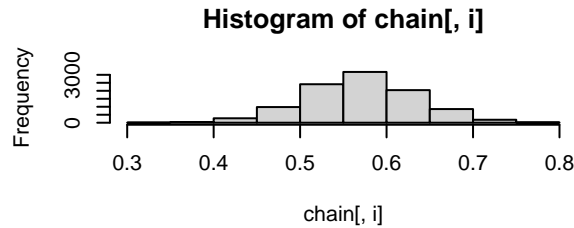
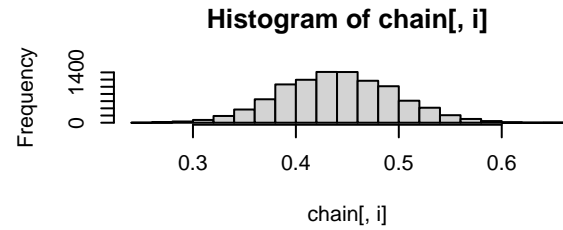
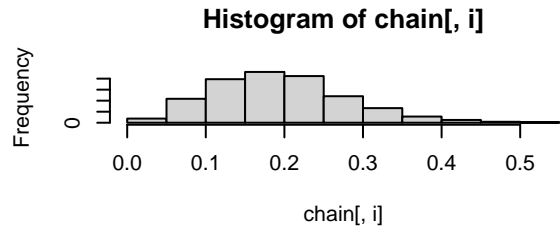
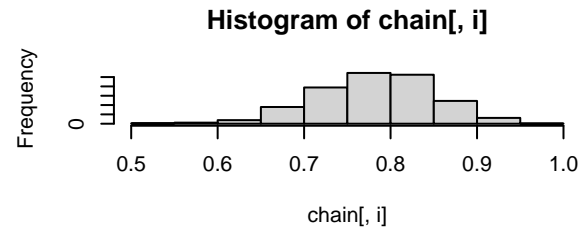
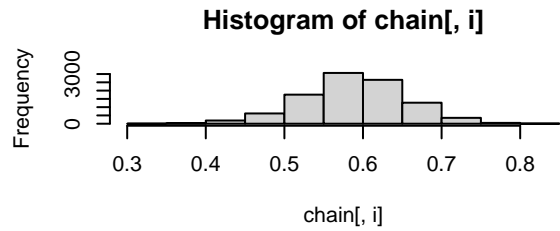
```

for (i in 1:6) plot(chain[,i], type='l')

```

```
for (i in 1:6) hist(chain[,i])
```



```
get_SE = function(x) # return variance of sample mean
{
  n = length(x)
  mu = sum(x) / n
  s_sq = sum((x - mu)^2) / ((n-1)*n) # notes 2 p 30
  return(sqrt(s_sq))
}
```

```
# point estimates of the 6 parameters
thetas = matrix(nrow=1,ncol=6)
thetas[1,] = colSums(chain) / length(chain[,1])
colnames(thetas) = c('theta_1', 'theta_2', 'theta_3', 'phi_1', 'phi_2', 'phi_3')
round(thetas, 4)
```

```
##      theta_1 theta_2 theta_3 phi_1 phi_2 phi_3
## [1,]  0.5894  0.7819  0.1946  0.4407  0.5696  0.3152
```

```
stderrs = matrix(nrow=1,ncol=6)
stderrs[1,] = c(get_SE(chain[,1]), get_SE(chain[,2]), get_SE(chain[,3]), get_SE(chain[,4]), get_SE(chain[,5]), get_SE(chain[,6]))
colnames(stderrs) = c('SE_theta_1', 'SE_theta_2', 'SE_theta_3', 'SE_phi_1', 'SE_phi_2', 'SE_phi_3')
round(stderrs, 6)
```

```
##      SE_theta_1 SE_theta_2 SE_theta_3 SE_phi_1 SE_phi_2 SE_phi_3
## [1,]  0.000652  0.000675  0.000846  0.00055  0.000636  0.000554
```

For the Odds Ratio, we do not know its distribution. But we can sample points from the posterior of its constituents θ_1 and ϕ_1 . We will obtain a point estimate and credible interval by simulating draws from the posterior distribution of θ_1, ϕ_1 , using them to calculate OR, and get its distribution. Then we will take the

quantiles of this empirical distribution to approximate the credible interval.

```
OR = function(t1, p1)
{
  return((t1 * (1-p1)) / (p1*(1-t1)))
}

chain = attempt_3_random_walk(n = 1000000)
chain = chain[1000:1000000,] # burn-in
ORs = OR(chain[,1], chain[,4])

# 95% credible interval of Odds Ratio
quantile(ORs, probs = c(0.025, 0.975))

##      2.5%      97.5%
## 0.9459062 3.7119857

# point estimate of Odds Ratio
mean(ORs)

## [1] 1.968251

par(mfrow=c(1,1))
hist(ORs)
```

Histogram of ORs

