HW 13

Problem 1

A novel test has:

- An α chance of detecting the cancer if the tumor is malignant,
- A .02 chance of falsely finding cancer if the tumor is benign.

Using the Bayes rule the updated $\pi(\theta = \text{cancer}|T)$ is

$$\frac{0.9\alpha}{0.9\alpha + 0.1 \times 0.02}, \text{ if } T \text{ is positive,}$$

$$\frac{0.9(1-\alpha)}{0.9(1-\alpha) + 0.1 \times 0.98}, \text{ if } T \text{ is negative.}$$

The quality-adjusted life expectancy given that T is positive is:

$$\begin{aligned}
u(a_1) &= 15.7\pi(\theta = \text{cancer}|T=1) + 33.8(1 - \pi(\theta = \text{cancer}|T=1)) \\
&= 33.8 - 18.1\pi(\theta = \text{cancer}|T=1), \\
u(a_2) &= 12.195\pi(\theta = \text{cancer}|T=1) + 21.62(1 - \pi(\theta = \text{cancer}|T=1)) \\
&= 21.62 - 9.425\pi(\theta = \text{cancer}|T=1), \\
u(a_3) &= 5.6\pi(\theta = \text{cancer}|T=1) + 34.8(1 - \pi(\theta = \text{cancer}|T=1)) \\
&= 34.8 - 29.2\pi(\theta = \text{cancer}|T=1).
\end{aligned}$$

Similarly the quality-adjusted life expectancy given that T is negative is:

$$\begin{aligned}
u(a_1) &= 33.8 - 18.1\pi(\theta = \text{cancer}|T=0), \\
u(a_2) &= 21.62 - 9.425\pi(\theta = \text{cancer}|T=0), \\
u(a_3) &= 34.8 - 29.2\pi(\theta = \text{cancer}|T=0).
\end{aligned}$$

First we notice that under our current assumption the life expectancy of a_2 the surgery way is always less than that of a_1 the radiotherapy way regardless of any medical test procedure. So it remains to compare a_1 and a_3. Solving the equation we know that when $\pi = 10/111$ the life expectancy of a_1 and a_3 are equal. This tells us that we need an α such that

$$\begin{aligned}
\frac{0.9\alpha}{0.9\alpha + 0.1 \times 0.02} &\geq 10/111 \\
\frac{0.9(1-\alpha)}{0.9(1-\alpha) + 0.1 \times 0.98} &< 10/111,
\end{aligned}$$

or the other way around to make sure that the test can affect the decision. Thus we have $\alpha \leq \frac{1}{4545}$ or $\alpha \geq \frac{4496}{4545}$.

If $\alpha \leq \frac{1}{4545}$, which means the test is stably bad at detecting real malignant tumor, in fact it is more likely that a malignant tumor is there if the test shows negative, then if the test is positive it 's better to do nothing and if the test is negative it is better to choose radiotherapy. On the other hand, if $\alpha \geq \frac{4496}{4545}$, which means the test is pretty solid, then if the test is positive then it's better to choose radiotherapy and if the test is negative it is better to do nothing. In both cases the test result can not be ignored and will affect the decision.

Problem 2

(1)

```
y0 <- c(rep(1, 557+38+4+3), rep(2, 427+27+1), rep(3, 87+1))
x <- c(rep(1, 557), rep(2, 38), rep(3, 4), rep(4, 3),
       rep(1, 427), rep(2, 27), rep(3, 1),
       rep(1, 87), rep(2, 1))

library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
y <- dummy(y0)
```

```
## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored
```

We are trying to study the population preference on their vote for the election. The data is collected by random calling phone lines and ask the person who picked the phone two questions: 1) Your preference 2) How many phone lines do you have in your house. The second question is necessary because if the household has several phone lines then the probability of getting sampled is multiplied by the number of phone lines. If we do not account this in our model then our survey is in favor of people who own more phone lines so the result is not fair. The following are technical details.

It will be too complicated to model the missing mechanism for x, the phone line number response, so we assume that it is missing completely at random. As a result it does not affect the inference of θ and we will not include it in the model. Also, we do not model the preference of households that do not have a phone line by assuming that not having phone line has nothing to do with their voting preference. This is also a missing completely at random assumption.

In general, the above assumptions mean that those households who did not respond to the "number of phone lines" question or do not own a phone share no different voting preference comparing to the population. As a result it is okay to model and estimate desired parameter ignoring those missing data.

Suppose the missing mechanism for y is known, that is, there is no parameter to be estimated for the missing mechanism. The complete model is

$$p(y, x, I|\theta, \phi) = p(y|\theta, x)p(I|x, y)p(x|\phi).$$

We've assume that y is missing completely at random so we have $p(I|x, y) = p(I|x)$. Then the complete model becomes

$$p(y, x, I|\theta, \phi) = p(y|\theta, x)p(I|x)p(x|\phi).$$

And thanks to this assumption the likelihood for observed data is just

$$\begin{aligned} p(y_{\text{obs}}, x, I|\theta, \phi) &= \int p(I|x)p(x|\phi)p(y_{\text{obs}}, y_{\text{mis}}|\theta, x)dy_{\text{mis}} \\ &= p(I|x)p(x|\phi)p(y_{\text{obs}}|\theta, x), \end{aligned}$$

where we set the probability of being sampled to be proportional to the number of phone lines

$$p(I_i = 1|x_i) \propto x_i$$

assuming that the total number of sampling candidate is finite. For the model of population number of phone lines we use Geometric distribution: for any $x_i \in 1, 2, 3, \ldots$

$$p(x_i|\phi) \propto (1 - \phi)^{x_i - 1}\phi.$$

And of course x_i are i.i.d. and so are I_i.

For the preference model, we set

$$y_i|x_i, \theta \sim y_i|\theta \sim \text{Multinomial}(1, \theta),$$

where $\theta = (\theta_1, \theta_2, \theta_3)$. Please note that we choose to believe that the phone lines has nothing to do with the voting preference so y does not depend on x. This does not mean that x contributes nothing to the estimation of θ. In fact, x plays an important role in the missing data mechanism thus it will appear in the complete likelihood and observed likelihood. Generally speaking, the way our model using the information in x reflects the fact that the preference sampled from households with multiple phone lines should be discredited a little bit since the probability of picking these households are higher.

To better construct the model we set

$$\begin{aligned} \alpha_1 &= \frac{\theta_1}{\theta_1 + \theta_2} \\ \alpha_2 &= 1 - \theta_3. \end{aligned}$$

In above parameterization α_1 is the probability of preferring Bush given that a preference is expressed and $alpha_2$ is the probability of expressing a preference. Then we relax these parameters to the logit scale in the following way

$$\beta_1 = \text{logit}(\alpha_1) \text{ and } \beta_2 = \text{logit}(\alpha_2),$$

where we assign a bivariate normal prior with fixed variance to β

$$\beta|\mu_1, \mu_2, \rho = \text{N}\left(\begin{pmatrix}\mu_1 \\ \mu 2\end{pmatrix}, \begin{pmatrix}1 & \rho \\ \rho & 1\end{pmatrix}\right).$$

Priors for hyperparameters (μ_1, μ_2, ρ) will be assigned later. The fixed variance prior for β is weakly informative but it is not designed to carry prior knowledge, instead it is mostly for convenience of computation.

Under the above hierarchical model, the posterior for (β, ϕ) is

$$p(\beta, \phi, \mu_1, \mu_2, \rho|x, y_{\text{obs}}, I) \propto p(\mu_1, \mu_2, \rho)p(\beta|\mu_1, \mu_2, \rho)p(\phi)\prod_{i=1}^{n} p(I_i|x_i)p(x_i|\phi)p(y_{\text{obs},i}|\beta, x_i).$$

Please note that we are using the β parameterization and β it is deterministically determined by θ. We will use a uniform prior for μ_1, μ_2, ρ and ϕ on feasible region and the resulting posterior is proper.

```r
library(mvtnorm)
library(mc2d)
```

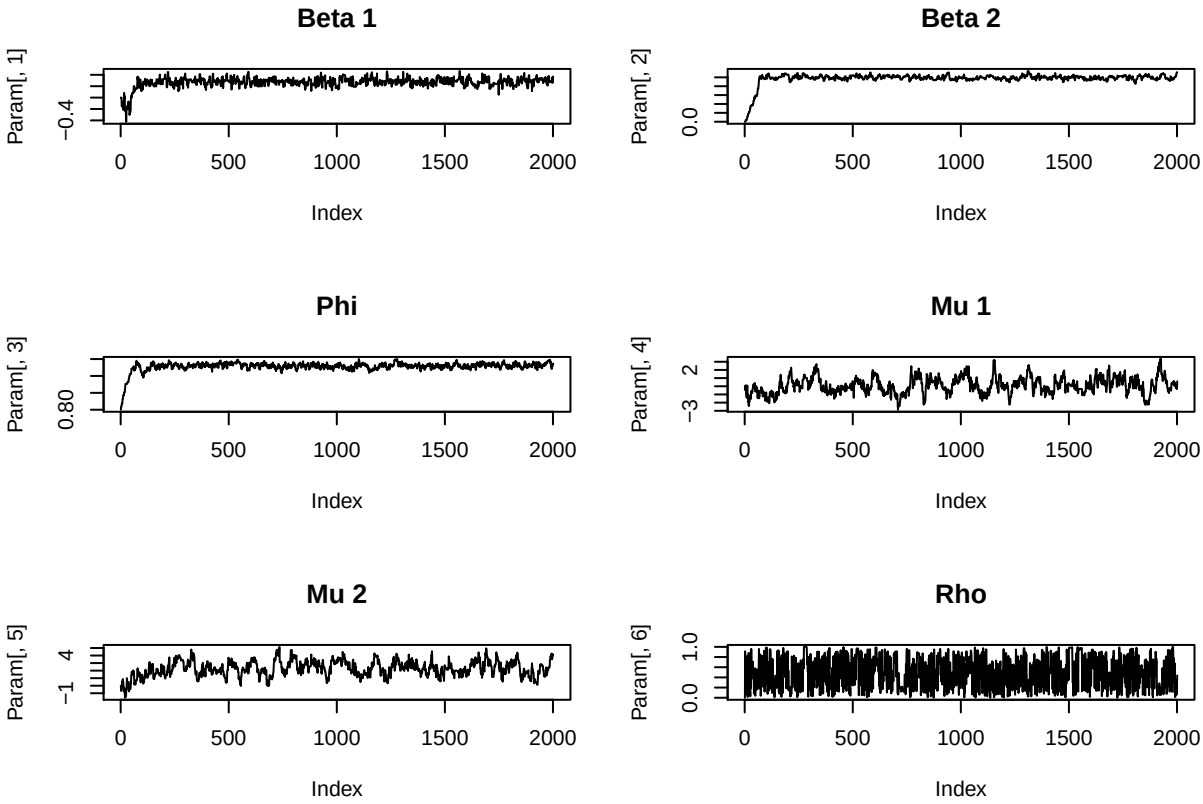
```
##
## Attaching package: 'mc2d'

## The following objects are masked from 'package:base':
##
##     pmax, pmin
```

```r
log_post <- function(beta, phi, mu, rho) {
  if (phi >0 & phi <=1 & rho <= 1 & rho >=-1) {
    sigma <- diag(c(1,1))
    sigma[1, 2] <- rho
    sigma[2, 1] <- sigma[1, 2]
    theta <- c(plogis(beta[1])*plogis(beta[2]),
               1-(1-plogis(beta[2]))-plogis(beta[1])*plogis(beta[2]),
               1-plogis(beta[2]))
    theta[theta < 0] <- 0 # fix numerical error like theta_2 = -3.41e-17
    dmvnorm(beta, mean=mu, sigma=sigma, log=T) + sum(log(x)) + sum(dgeom(x-1, phi, log=T)) +
      sum(dmultinomial(y, prob=theta, log=T))
  } else {
    -Inf
  }
}

init <- c(0, 0, 0.8, 0, 0, 0)
iter <- 2e3

Param <- matrix(0, nrow=iter+1, ncol=length(init))
Param[1, ] <- init

for (i in 1:iter) {
  cand_beta <- rnorm(2, mean=Param[i, 1:2], sd=.1)
  log_u <- log(runif(1))
  obj <- log_post(cand_beta, Param[i, 3], Param[i, 4:5], Param[i, 6]) -
         log_post(Param[i, 1:2], Param[i, 3], Param[i, 4:5], Param[i, 6])
  if (log_u <= obj) {
    Param[i+1, 1:2] <- cand_beta
  } else {
    Param[i+1, 1:2] <- Param[i, 1:2]
  }

  cand_phi <- runif(1, min=max(0, Param[i, 3]-.01), max=min(1, Param[i, 3] + .01))
  log_u <- log(runif(1))
  obj <- log_post(Param[i+1, 1:2], cand_phi, Param[i, 4:5], Param[i, 6]) -
         log_post(Param[i+1, 1:2], Param[i, 3], Param[i, 4:5], Param[i, 6]) +
         dunif(Param[i, 3], min=max(0, cand_phi-.01),
               max=min(1, cand_phi + .01), log=T) -
         dunif(cand_phi, min=max(0, Param[i, 3]-.01),
                max=min(1, Param[i, 3] + .01), log=T)
   if (log_u <= obj) {
    Param[i+1, 3] <- cand_phi
```

```
  } else {
    Param[i+1, 3] <- Param[i, 3]
  }

  cand_mu <- rnorm(2, mean=Param[i, 4:5], sd=.5)
  log_u <- log(runif(1))
  obj <- log_post(Param[i+1, 1:2], Param[i+1, 3], cand_mu, Param[i, 6]) -
         log_post(Param[i+1, 1:2], Param[i+1, 3], Param[i, 4:5], Param[i, 6])
  if (log_u <= obj) {
    Param[i+1, 4:5] <- cand_mu
  } else {
    Param[i+1, 4:5] <- Param[i, 4:5]
  }

  cand_rho <- runif(1)
  log_u <- log(runif(1))
  obj <- log_post(Param[i+1, 1:2], Param[i+1, 3], Param[i+1, 4:5], cand_rho) -
         log_post(Param[i+1, 1:2], Param[i+1, 3], Param[i+1, 4:5], Param[i, 6])
  if (log_u <= obj) {
    Param[i+1, 6] <- cand_rho
  } else {
    Param[i+1, 6] <- Param[i, 6]
  }

}
```

Let's take a look at the path

```
par(mfrow=c(3,2))
plot(Param[, 1], type='l', main='Beta 1')
plot(Param[, 2], type='l', main='Beta 2')

plot(Param[, 3], type='l', main='Phi')
plot(Param[, 4], type='l', main='Mu 1')

plot(Param[, 5], type='l', main='Mu 2')
plot(Param[, 6], type='l', main='Rho')
```



Although we did not use multiple chains to check mixing status, the above plots are sufficient to tell that the chain converges.

The posterior mean of parameters are

```
colMeans(Param[round(.5*iter):iter+1, ])
```

```
## [1] 0.2838107 2.4805586 0.9299390 0.2639537 2.3512749 0.4943847
```

To be specific, the posterior mean for θ is

```
beta <- colMeans(Param[round(.5*iter):iter+1, 1:2])
theta <- c(plogis(beta[1])*plogis(beta[2]),
           1-(1-plogis(beta[2]))-plogis(beta[1])*plogis(beta[2]),
           1-plogis(beta[2]))
theta
```

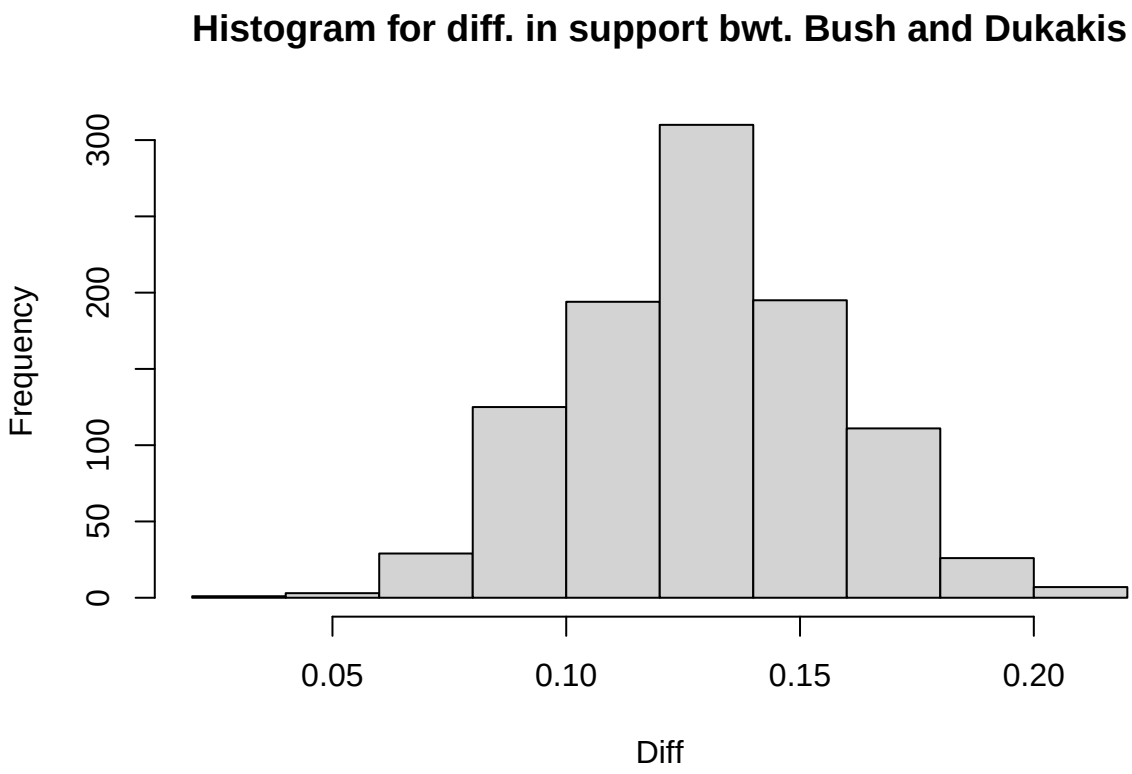
```
## [1] 0.52642068 0.39634694 0.07723238
```

And the posterior mean for ϕ is

```
mean(Param[round(.5*iter):iter+1, 3])
```

```
## [1] 0.929939
```

```r
theta <- cbind(plogis(Param[, 1])*plogis(Param[, 2]),
               1-(1-plogis(Param[, 2]))-plogis(Param[, 1])*plogis(Param[, 2]),
               1-plogis(Param[, 2]))
hist( (theta[, 1] - theta[,2])[round(.5*iter):iter+1],
  main='Histogram for diff. in support bwt. Bush and Dukakis',
  xlab='Diff')
```



Next we do a posterior predictive check for T_1 = proportion of households that support Bush, T_2 = proportion of households that do not express preference and T_3 = average household phone lines. Under our assumptions it is safe to do posterior predictive check using simulated complete data to compare with observed data since their likelihood functions are the same.

```r
theta_S <- theta[round(.5*iter):iter+1, ]
phi_S <- Param[round(.5*iter):iter+1, 3]
S <- dim(theta_S)[1]

T_rep <- matrix(0, nrow=S, ncol=3)
T_0 <- c(mean(y0 == 1), mean(y0 == 3), mean(x))
for (i in 1:S) {
  y_rep <- rmultinomial(length(y0), size=1, prob=theta_S[i, ])
  x_rep <- rgeom(length(x), prob=phi_S[i])

  T_rep[i, ] <- c(mean(y_rep[, 1] == 1), mean(y_rep[, 3] == 1), mean(x_rep))
}
```

Estimated p-values for T_1, T_2 and T_3 are

```r
mean(T_rep[, 1] >= T_0[1])
```

```
## [1] 0.5184815
```

```r
mean(T_rep[, 2] >= T_0[2])
```

```
## [1] 0.5314685
```

```r
mean(T_rep[, 3] >= T_0[3])
```

```
## [1] 0
```

(2)

If we further consider the fact that there could be more than one adult in a household, we need to further assume that the number of adults in a household does not affect the preference of these people in the household. Also, our estimand is the preference among individuals not households.

Denote the number of adults in a household by z. The complete model will be the same except the model $p(I_i = 1|x_i)$ is now $p(I_i = 1|x_i, z_i)$ and

$$p(I_i = 1|x_i, z_i) \propto \frac{x_i}{z_i}$$

given that the total number of sampling candidates is finite. We choose that $z_i - 1$ follows a poisson distribution with parameter λ:

$$p(z_i - 1|\lambda) = \frac{\lambda^{z_i - 1} e^{-\lambda}}{(z_i - 1)!}.$$

The prior for λ will be uniform on $(0, +\infty)$.

The posterior is

$$p(\beta, \phi, \lambda, \mu_1, \mu_2, \rho|x, y_{\text{obs}}, I) \propto p(\mu_1, \mu_2, \rho)p(\beta|\mu_1, \mu_2, \rho)p(\phi)p(\lambda) \prod_{i=1}^{n} p(I_i|x_i, z_i)p(x_i|\phi)p(z_i|\lambda)p(y_{\text{obs},i}|\beta, x_i, z_i).$$

Please note that our model assumes that the number of phone lines is independent with the number of adults in a household. This is reasonable if we are not considering cell phones. Also, the observed likelihood is written as $p(y_{\text{obs},i}|\beta, x_i, z_i)$ but in fact it is equals to $p(y_{\text{obs},i}|\beta)$ since we choose that y does not depend on x or z.

```r
dat_raw <- c(124, 3, 0, 2, 2,
134, 2, 0, 0, 0,
32, 0, 0, 0, 1,
 332, 21, 3, 0, 5,
229, 15, 0, 0, 3,
47, 0, 0, 0, 6,
 71, 9, 1, 0, 0,
47, 7, 1, 0, 0,
4, 1, 0, 0, 0,
 23, 4, 0, 1, 0,
11, 3, 0, 0, 0,
```

```
3, 0, 0, 0, 0,
 3, 0, 0, 0, 0,
4, 0, 0, 0, 0,
1, 0, 0, 0, 0,
 1, 0, 0, 0, 0,
1, 0, 0, 0, 0,
0, 0, 0, 0, 0,
 2, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
 1, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0)
```

The data is not in a convenient format so let's first do some reshaping to get the x, y and z is vector form.

```
dat <- t(matrix(dat_raw, nrow=5, ncol=24))
dat <- dat[, -5]
dat <- cbind(dat, rep(seq(1:8), each=3))

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
z0 <- dat %>% as.data.frame() %>%
  group_by(V5) %>%
  summarise(sum(V1, V2, V3, V4))

y00 <- dat %>% rowSums()
z <- rep(seq(1:8), as.numeric(z0[,2][[1]]))
y0 <- rep(rep(seq(1:3), 8), dat %>% rowSums())
x <- rep(rep(seq(1:4), 24), c(t(dat[, -5])))

library(dummies)
y <- dummy(y0)
```

```
## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored
```

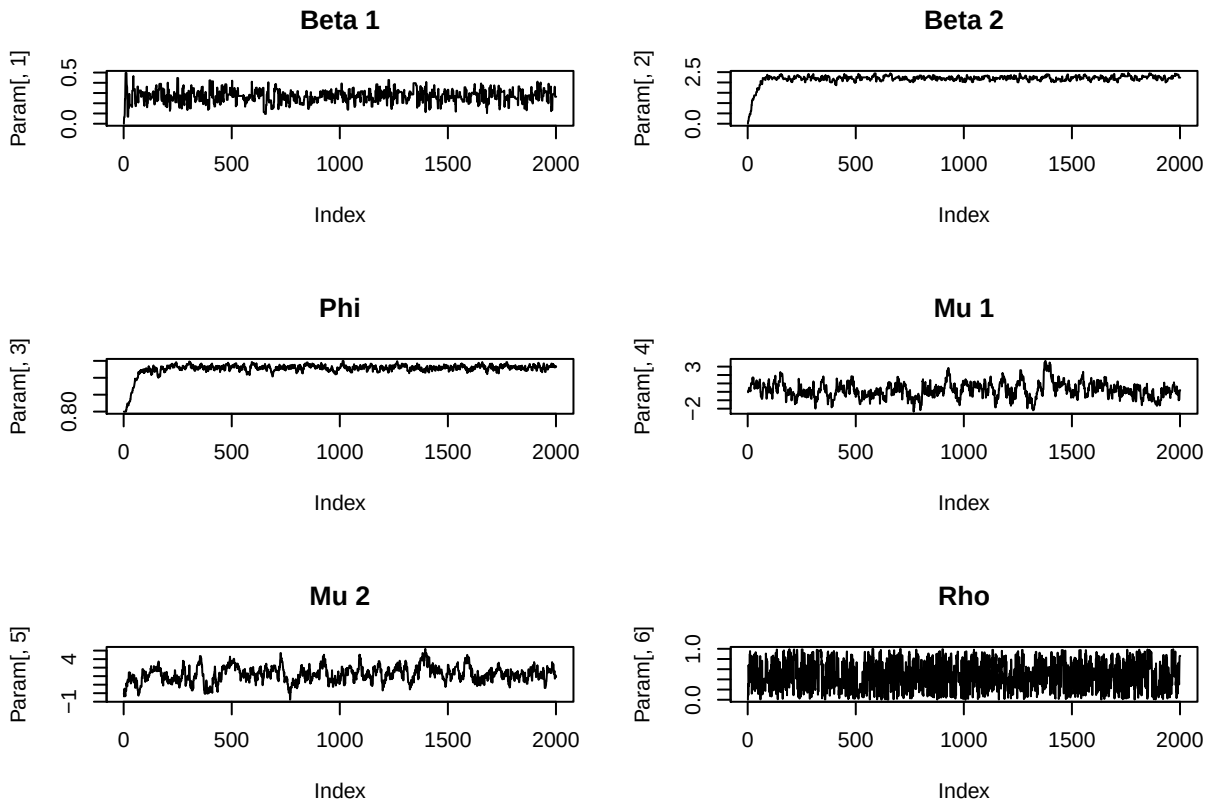
Now we are ready to sample from the posterior.

```r
log_post <- function(beta, phi, mu, rho, log_lambda) {
  if (phi >0 & phi <=1 & rho <= 1 & rho >=-1 ) {
    lambda <- exp(log_lambda)
    sigma <- diag(c(1,1))
    sigma[1, 2] <- rho
    sigma[2, 1] <- sigma[1, 2]
    theta <- c(plogis(beta[1])*plogis(beta[2]),
               1-(1-plogis(beta[2]))-plogis(beta[1])*plogis(beta[2]),
               1-plogis(beta[2]))
    theta[theta < 0] <- 0 # fix numerical error like theta_2 = -3.41e-17
    dmvnorm(beta, mean=mu, sigma=sigma, log=T) + sum(log(x/z)) + sum(dgeom(x-1, phi, log=T)) +
      sum(dmultinomial(y, prob=theta, log=T)) + sum(dpois(z-1, lambda, log=T))
  } else {
    -Inf
  }
}

init <- c(0, 0, 0.8, 0, 0, 0, 0)
iter <- 2e3

Param <- matrix(0, nrow=iter+1, ncol=length(init))
Param[1, ] <- init

for (i in 1:iter) {
  cand_beta <- rnorm(2, mean=Param[i, 1:2], sd=.1)
  log_u <- log(runif(1))
  obj <- log_post(cand_beta, Param[i, 3],
                  Param[i, 4:5], Param[i, 6], Param[i, 7]) -
         log_post(Param[i, 1:2], Param[i, 3],
                  Param[i, 4:5], Param[i, 6], Param[i, 7])
  if (log_u <= obj) {
    Param[i+1, 1:2] <- cand_beta
  } else {
    Param[i+1, 1:2] <- Param[i, 1:2]
  }

  cand_phi <- runif(1, min=max(0, Param[i, 3]-.01), max=min(1, Param[i, 3] + .01))
  log_u <- log(runif(1))
  obj <- log_post(Param[i+1, 1:2], cand_phi,
                  Param[i, 4:5], Param[i, 6], Param[i, 7]) -
         log_post(Param[i+1, 1:2], Param[i, 3],
                  Param[i, 4:5], Param[i, 6], Param[i, 7]) +
         dunif(Param[i, 3], min=max(0, cand_phi-.01),
               max=min(1, cand_phi + .01), log=T) -
         dunif(cand_phi, min=max(0, Param[i, 3]-.01),
               max=min(1, Param[i, 3] + .01), log=T)
   if (log_u <= obj) {
    Param[i+1, 3] <- cand_phi
  } else {
    Param[i+1, 3] <- Param[i, 3]
  }

  cand_mu <- rnorm(2, mean=Param[i, 4:5], sd=.5)
```

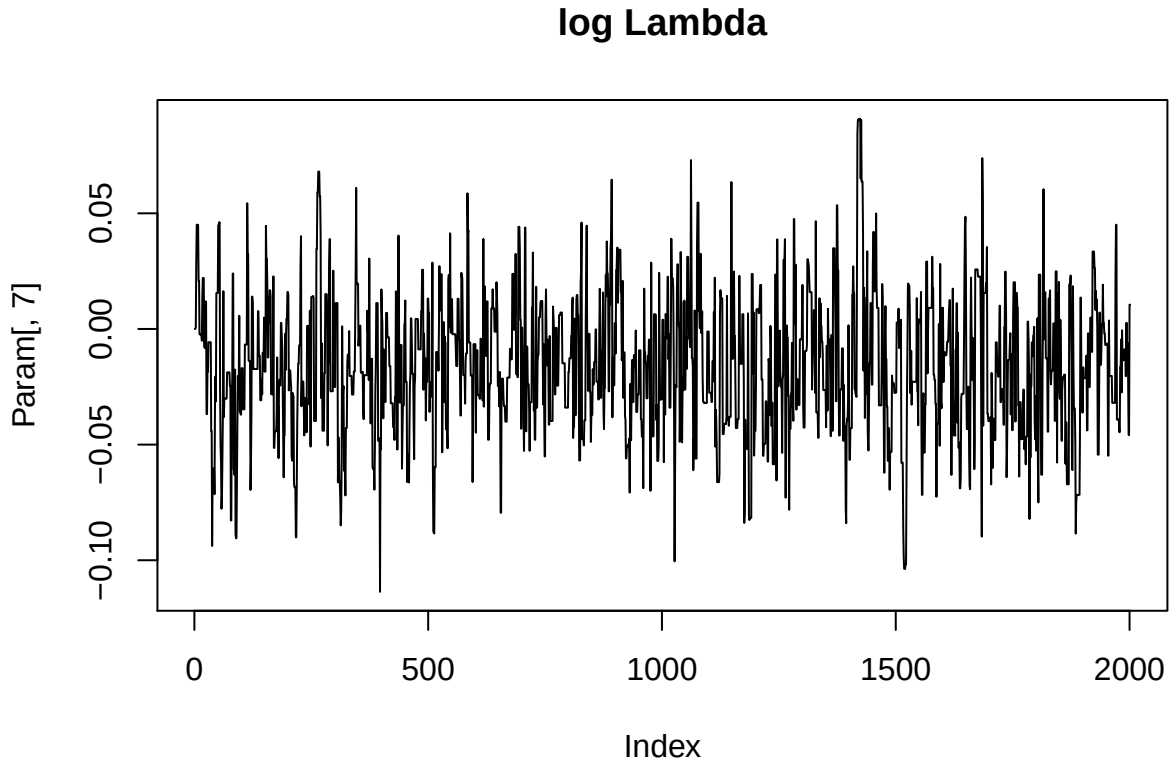
```
  log_u <- log(runif(1))
  obj <- log_post(Param[i+1, 1:2], Param[i+1, 3],
                  cand_mu, Param[i, 6], Param[i, 7]) -
         log_post(Param[i+1, 1:2], Param[i+1, 3],
                  Param[i, 4:5], Param[i, 6], Param[i, 7])
  if (log_u <= obj) {
    Param[i+1, 4:5] <- cand_mu
  } else {
    Param[i+1, 4:5] <- Param[i, 4:5]
  }

  cand_rho <- runif(1)
  log_u <- log(runif(1))
  obj <- log_post(Param[i+1, 1:2],  Param[i+1, 3],
                  Param[i+1, 4:5], cand_rho, Param[i, 7])   -
         log_post(Param[i+1, 1:2], Param[i+1, 3],
                  Param[i+1, 4:5], Param[i, 6], Param[i, 7])
  if (log_u <= obj) {
    Param[i+1, 6] <- cand_rho
  } else {
    Param[i+1, 6] <- Param[i, 6]
  }

  cand_lambda <- rnorm(1, mean=Param[i, 7], sd=.05)
  log_u <- log(runif(1))
  obj <- log_post(Param[i+1, 1:2],  Param[i+1, 3],
                  Param[i+1, 4:5], Param[i+1, 6], cand_lambda)   -
         log_post(Param[i+1, 1:2], Param[i+1, 3],
                  Param[i+1, 4:5], Param[i+1, 6], Param[i, 7])
  if (log_u <= obj) {
    Param[i+1, 7] <- cand_lambda
  } else {
    Param[i+1, 7] <- Param[i, 7]
  }

}
```

Let's take a look at the path

```
par(mfrow=c(3,2))
plot(Param[, 1], type='l', main='Beta 1')
plot(Param[, 2], type='l', main='Beta 2')

plot(Param[, 3], type='l', main='Phi')
plot(Param[, 4], type='l', main='Mu 1')

plot(Param[, 5], type='l', main='Mu 2')
plot(Param[, 6], type='l', main='Rho')
```




```
plot(Param[, 7], type='l', main='log Lambda')
```



The above plots are sufficient to tell that the chain converges.

The posterior mean of parameters are

```
colMeans(Param[round(.5*iter):iter+1, ])
```

```
## [1]  0.27195213  2.21717758  0.93009794  0.31825256  2.39226977  0.49478323
## [7] -0.01606228
```

To be specific, the posterior mean for θ is

```
beta <- colMeans(Param[round(.5*iter):iter+1, 1:2])
theta <- c(plogis(beta[1])*plogis(beta[2]),
             1-(1-plogis(beta[2]))-plogis(beta[1])*plogis(beta[2]),
             1-plogis(beta[2]))
theta
```

```
## [1] 0.51182600 0.38995549 0.09821851
```

And the posterior mean for ϕ and λ is

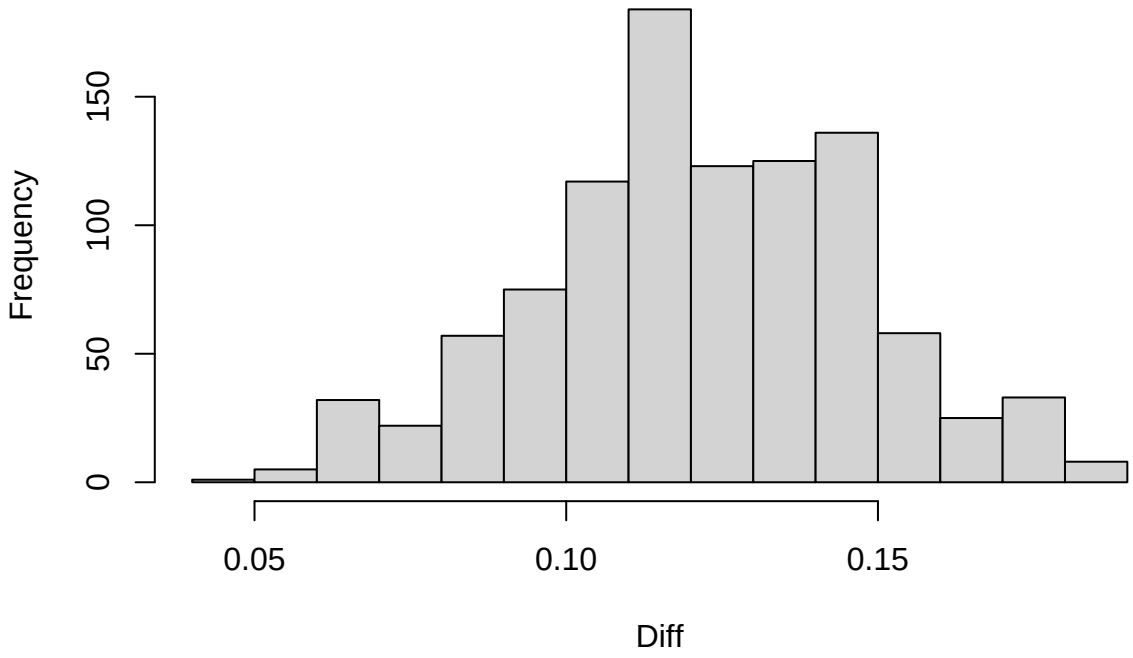
```
mean(Param[round(.5*iter):iter+1, 3])
```

```
## [1] 0.9300979
```

```
mean(exp(Param[round(.5*iter):iter+1, 7]))
```

```
## [1] 0.9845633
```

```
theta <- cbind(plogis(Param[, 1])*plogis(Param[, 2]),
             1-(1-plogis(Param[, 2]))-plogis(Param[, 1])*plogis(Param[, 2]),
             1-plogis(Param[, 2]))
hist( (theta[, 1] - theta[,2])[round(.5*iter):iter+1],
  main='Histogram for diff. in support bwt. Bush and Dukakis',
  xlab='Diff')
```



Next we do a posterior predictive check for T_1 = proportion of households that support Bush, T_2 = proportion of households that do not express preference and T_3 = average phone lines and T_4 = average adult numbers. Under our assumptions it is safe to do posterior predictive check using simulated complete data to compare with observed data since their likelihood functions are the same.

```
theta_S <- theta[round(.5*iter):iter+1, ]
phi_S <- Param[round(.5*iter):iter+1, 3]
lambda_S <- exp(Param[round(.5*iter):iter+1, 7])
S <- dim(theta_S)[1]

T_rep <- matrix(0, nrow=S, ncol=4)
T_0 <- c(mean(y0 == 1), mean(y0 == 3), mean(x), mean(z))
for (i in 1:S) {
  y_rep <- rmultinomial(length(y0), size=1, prob=theta_S[i, ])
  x_rep <- rgeom(length(x), prob=phi_S[i])
  z_rep <- rpois(length(z), lambda=lambda_S[i])
  T_rep[i, ] <- c(mean(y_rep[, 1] == 1), mean(y_rep[, 3] == 1),
                  mean(x_rep), mean(z_rep))
}
```

Estimated p-values for T_1, T_2 and T_3 are

```
mean(T_rep[, 1] >= T_0[1])
```

```
## [1] 0.5614386
```

```
mean(T_rep[, 2] >= T_0[2])
```

```
## [1] 0.4465534
```

```r
mean(T_rep[, 3] >= T_0[3])
```

```
## [1] 0
```

```r
mean(T_rep[, 4] >= T_0[4])
```

```
## [1] 0
```

Although the posterior means reflect what the data says, the posterior predictive check results are not promising for the estimation of θ.