# STA 623 HW7

*Lingyun Shao*

*2018/11/14*

## 1. Model

Including $c$ and $d$ in the model, the parameters and data are modeled as below:

$$\begin{aligned} Y|S,d &\sim \text{Beta}(1+dS, 1+d(1-S)) \\ X|\theta &\sim \text{Normal}(\text{logit}(\theta), \sigma) \\ S|\pi &\sim \text{Bernoulli}(\pi) \\ \pi|\theta,c &\sim \text{Beta}(1+c\theta, 1+c(1-\theta)) \\ \theta|a,b &\sim \text{Beta}(a,b) \\ a &\sim \text{Gamma}(\alpha,\beta) \\ b &\sim \text{Gamma}(\alpha,\beta) \\ c &\sim \text{Exp}(1) \\ d &\sim \text{Gamma}(2,2) \end{aligned}$$

We are interested in probability $Pr(S=1|X=x, Y=y)$, which can be derived by $\frac{P(S=1,X=x,Y=y)}{\sum_{i=0,1} P(S=i,X=x,Y=y)}$. We can get the marginal by integrating out $(\pi,\theta,a,b,c,d)$, or just use MCMC samples $(\pi^{(t)},\theta^{(t)},a^{(t)},b^{(t)},c^{(t)},d^{(t)})$ from stan and do the average.

From the instruction of Lab session 2, we know the joint density can be written as:

$$\begin{aligned} &P(S=1,X,Y,\theta,\pi,a,b,c,d) \\ =&p(Y|S=1,d)P(S|\pi)p(X|\theta)p(\pi,\theta,a,b,c,d) \end{aligned}$$

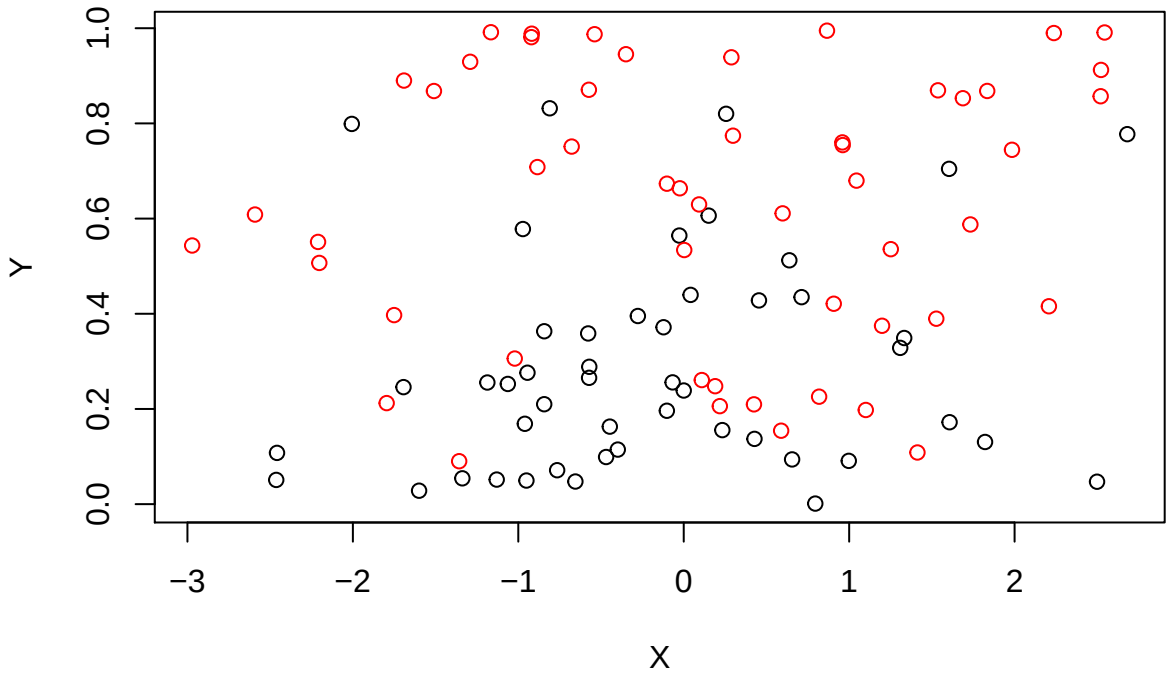
## 2. Modifications and Results

```
data{
  int<lower=1> n;
  vector[n] Y;
  vector[n] X;
  real<lower=0> sigma;
  real<lower=0> alpha;
  real<lower=0> beta;
  real<lower=0> lambda; // Modified
  real<lower=0> alpha_d; // Modified
  real<lower=0> beta_d; // Modified
}

parameters{
  real<lower=0> a;
  real<lower=0> b;
```

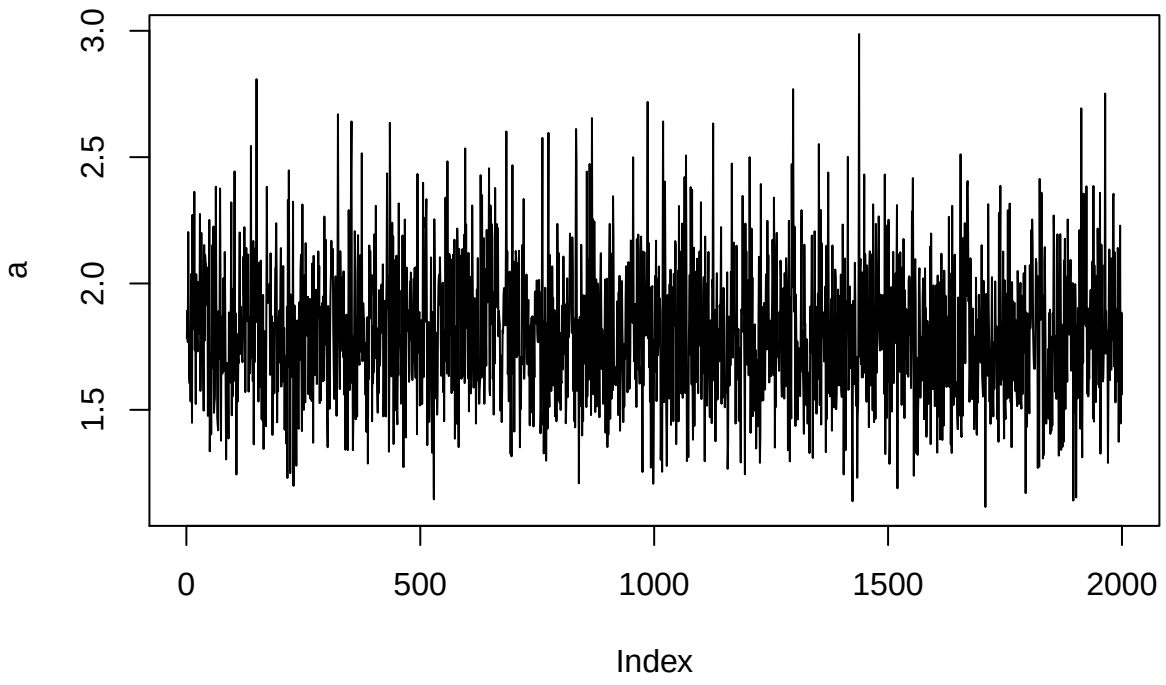
```
  real<lower=0> c; // Modified
  real<lower=0> d; // Modified
  vector<lower=0, upper=1>[n] theta;
  vector<lower=0, upper=1>[n] Pi;
  // Additional independent chain (pi,theta)
  // to estimate integrals.

}

model{

  a ~ gamma(alpha,beta); // target += gamma_lpdf(a | alpha,beta);
  b ~ gamma(alpha,beta); // target += gamma_lpdf(b | alpha,beta);
  c ~ exponential(lambda); // Modified
  d ~ gamma(alpha_d,beta_d); // Modified

  for(i in 1:n){

    theta[i] ~ beta(a,b); // target += beta_lpdf(theta[i] | a,b);

    Pi[i] ~ beta(1 + c*theta[i], 1 + c*(1-theta[i])); // Modified

    X[i] ~ normal(logit(theta[i]), sigma);
    // target += normal_lpdf(logit(theta[i]), sigma);

    target += log_sum_exp(log(Pi[i]) + beta_lpdf(Y[i] | 1+d, 1), log(1-Pi[i]) + beta_lpdf(Y[i] | 1, 1+d
  }

}

generated quantities{
  real<lower=0, upper=1> Pi_s;
  real<lower=0, upper=1> theta_s;

  theta_s = beta_rng(a,b);
  Pi_s = beta_rng(1 + c*theta_s, 1 + c*(1-theta_s)); // Modified
}
```

```
####################
# MCMC HW7         #
####################

# Model
# -----
#
# Modified : added c and d
# theta ~ beta(a,b)                                    : Level of chemical M
# X | theta ~ normal(logit(theta), sigma)              : Test X, measuring level of M
# pi | c, theta ~ Beta(1 + c*theta, 1 + c*(1 - theta)) : Probability that patient has sickness, high le
# Sick | pi ~ Bernoulli(pi)                            : Patient having the sickness
# Y | d, Sick ~ Beta(1 + d*Sick, 1 + d(1 - Sick))      : Test Y, sick patients score tend to be much hi
# a, b ~ Gamma(2,1)                                    : Population parameters of levels of M
# c ~ Exp(1)
# d ~ Gamma(2,2)
```

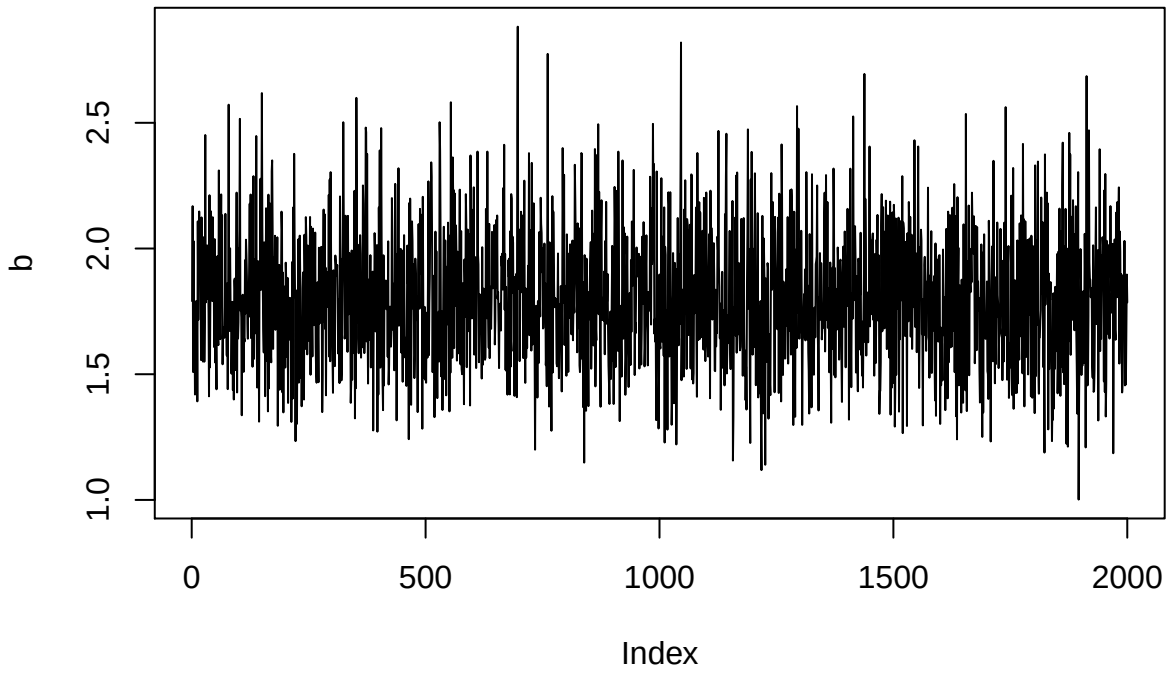
```r
library(rstan)


# Prior and known parameters
sigma <- 0.5
alpha <- 2
beta <- 1
lambda = 1 # Modified
alpha_d = 2 # Modified
beta_d = 2 # Modified

# Create data
set.seed(2112)
n <- 100

# Latent theta, probability of illness, and illness indicator.
th0 <- rbeta(n,2,2)
# Modified : The original code is commented, I'm not sure whether this code is wrong or not
# But judging from the formula in Lab 2 notes, I think the parameters should not be multiplied by 2
# And I also set 'c' here to be its expectation in a exponential distribution, that is 1.
# pi0 <- rbeta(n,2*(1 + 2*th0), 2*(1 + 2*(1-th0)))
pi0 = rbeta(n,(1 + 1*th0), (1 + 1*(1-th0)))
S0 <- rbinom(n,1,pi0)

# Test results
X <- rnorm(n, log(th0)-log(1-th0), sigma)
Y <- rbeta(n, 1 + S0, 1 + (1 - S0))

# A plot. (remember we do not know the health of the data, only X,Y)
plot(X,Y,col=S0+1)
```



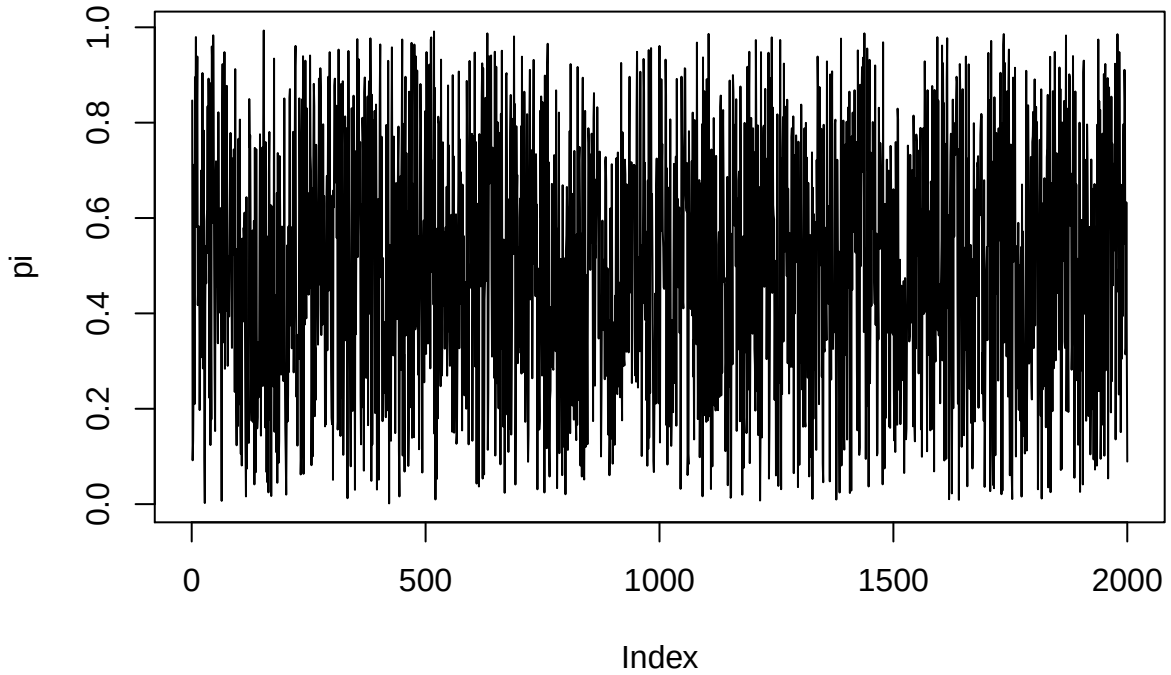
```r
# Data format for Stan.
stan_data <- list(n = n, Y = Y, X = X, sigma = sigma, alpha = alpha, beta = beta,
                  lambda = lambda, alpha_d = alpha_d, beta_d = beta_d)

# Execute this line and find the file (only once)
# if(!exists('stan_file')){ stan_file <- file.choose() } # Modified : added quote
stan_file = 'HW7.stan'

T <- 2000
B <- 500
# This is the Stan execution, may take a while.
fit <- stan(file = stan_file, data = stan_data,
            iter = B+T, warmup = B, chains=1) # Modified : fixed warmup to B

# Extract MCMC chains, we will need pi and theta.
draws <- extract(fit, pars = c("a","b",'c', 'd', "Pi_s","theta_s"))  # Modified


# Some trace plots
plot(draws$a, type="l", ylab="a")
```



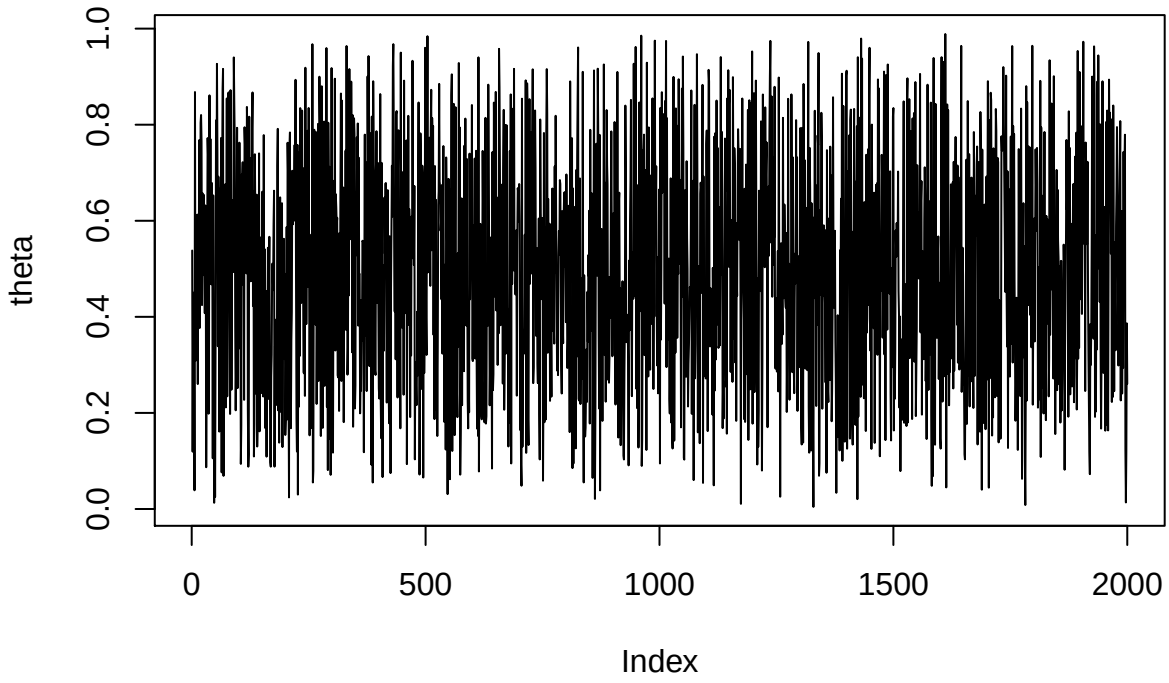

```r
plot(draws$b, type="l", ylab="b")
```



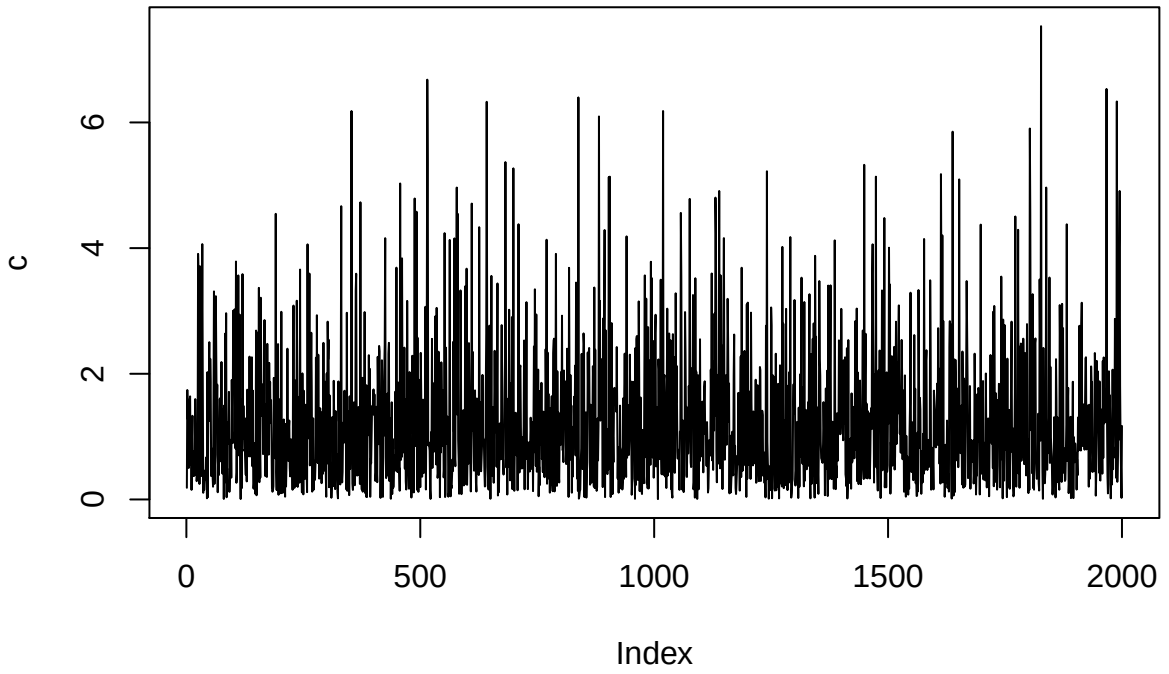

```
plot(draws$Pi_s, type="l", ylab="pi")
```



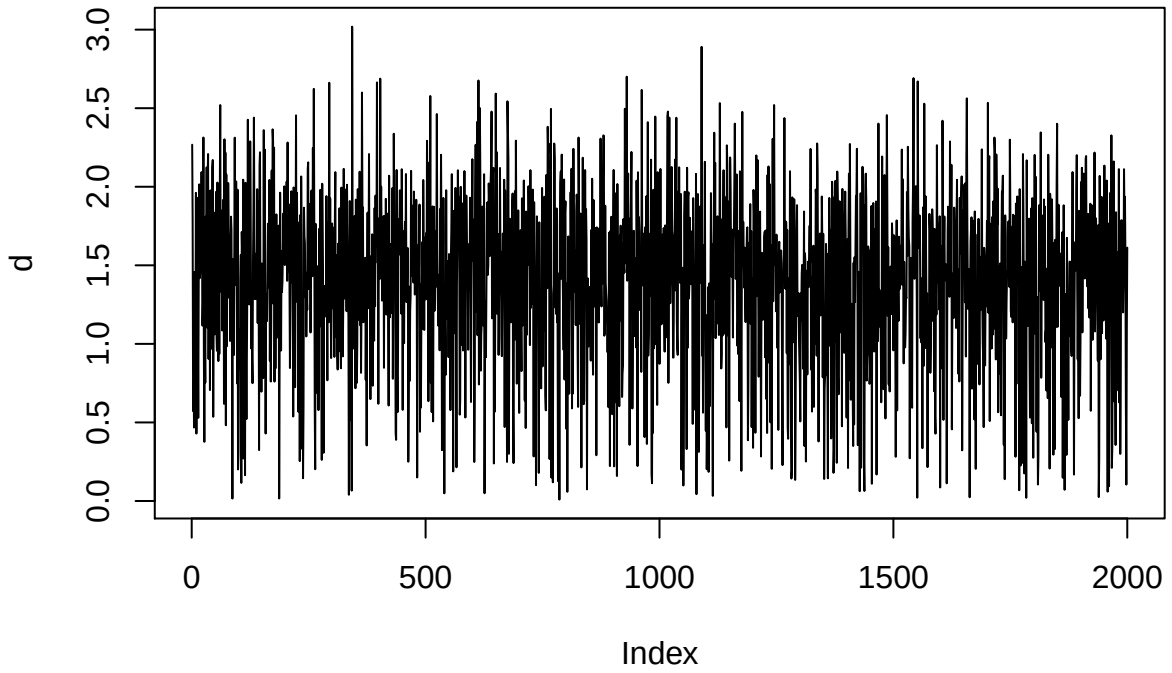

```
plot(draws$theta_s, type="l", ylab="theta") # Modified : fixed label to theta
```




```
plot(draws$c, type="l", ylab="c") # Modified
```

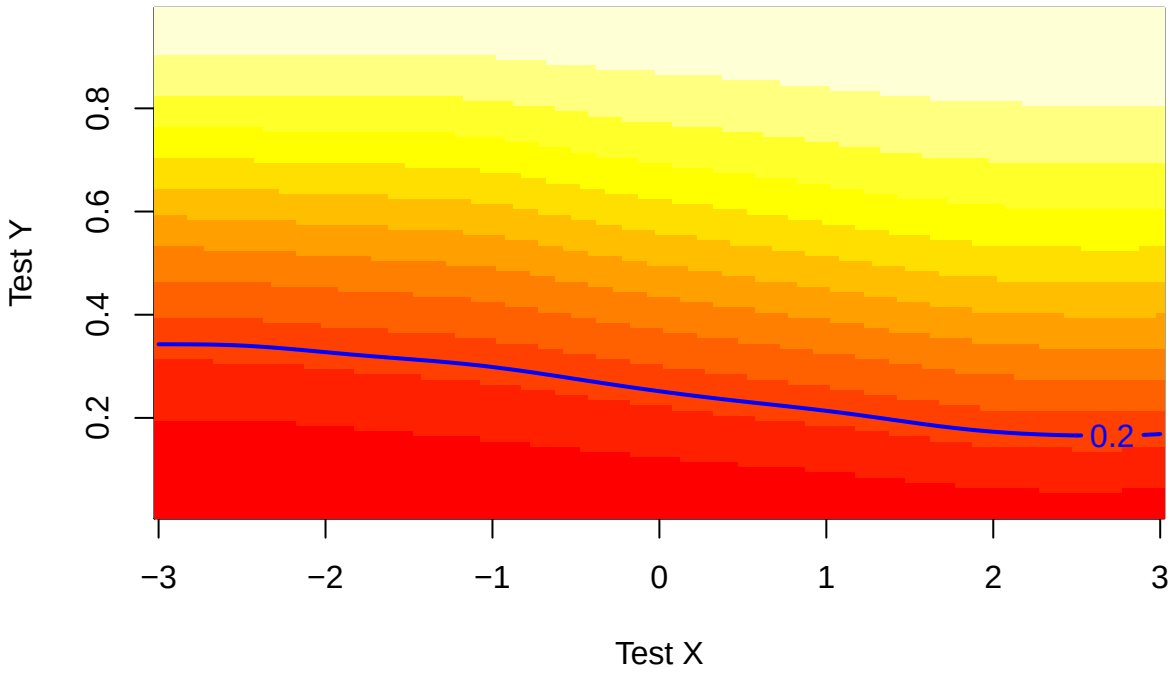



```
plot(draws$d, type="l", ylab="d") # Modified
```

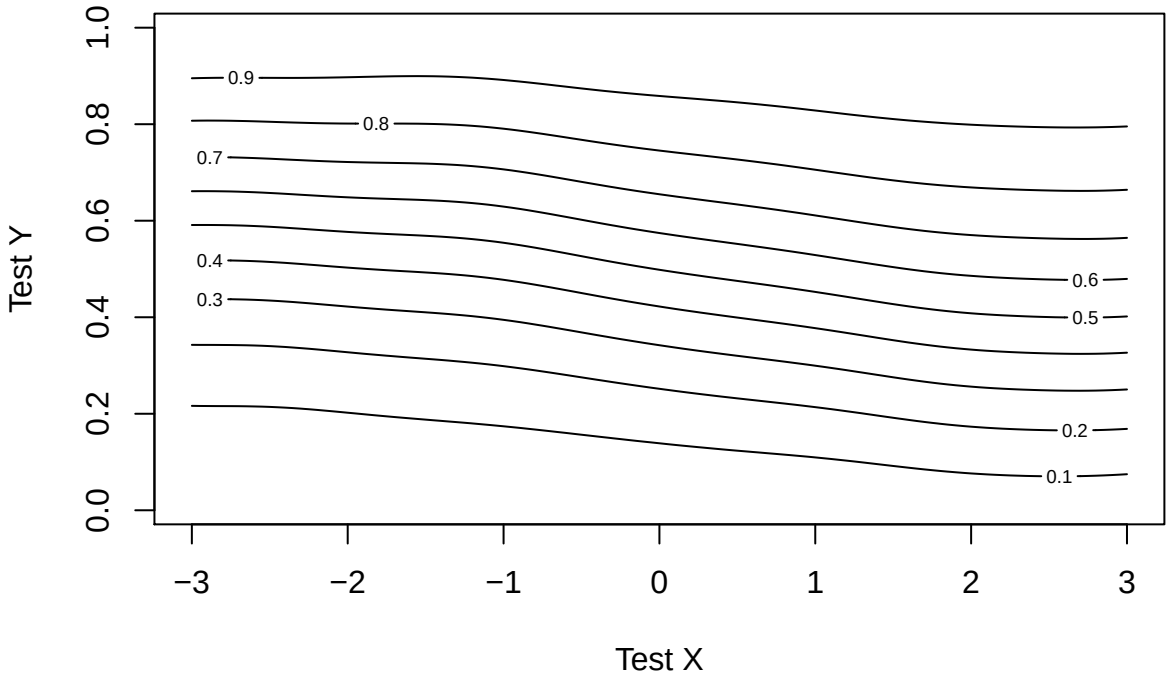



```
## EXERCISE 3: here

# Since we are using samples from stan, we don't need to sample in R


##

# Short names for mcmc chains. Erase these two lines if using EXERCISE 3 code.
theta <- draws$theta_s
pi <- draws$Pi_s
c <- draws$c # Modified
d <- draws$d # Modified

# Logit of theta's
logit_theta <- log(theta)-log(1-theta)

# Define the Monte Carlo approximation of gamma(x,y)
gamma <- function(x,y){

  numerator <- dbeta(y,1+d,1) * pi * dnorm(x, logit_theta, sigma) # Modified
  denominator <- (dbeta(y,1+d,1)* pi + dbeta(y,1,1+d) * (1-pi)) * dnorm(x, logit_theta, sigma) # Modifi

  # Probability value
  mean(numerator)/mean(denominator)

}

# Now let's set a range of interest for both test values.
seq_x <- seq(-3,3,0.05)
seq_y <- seq(0.01,0.99,0.01)
gamma_matrix <- array(0,c(length(seq_x),length(seq_y)))
```

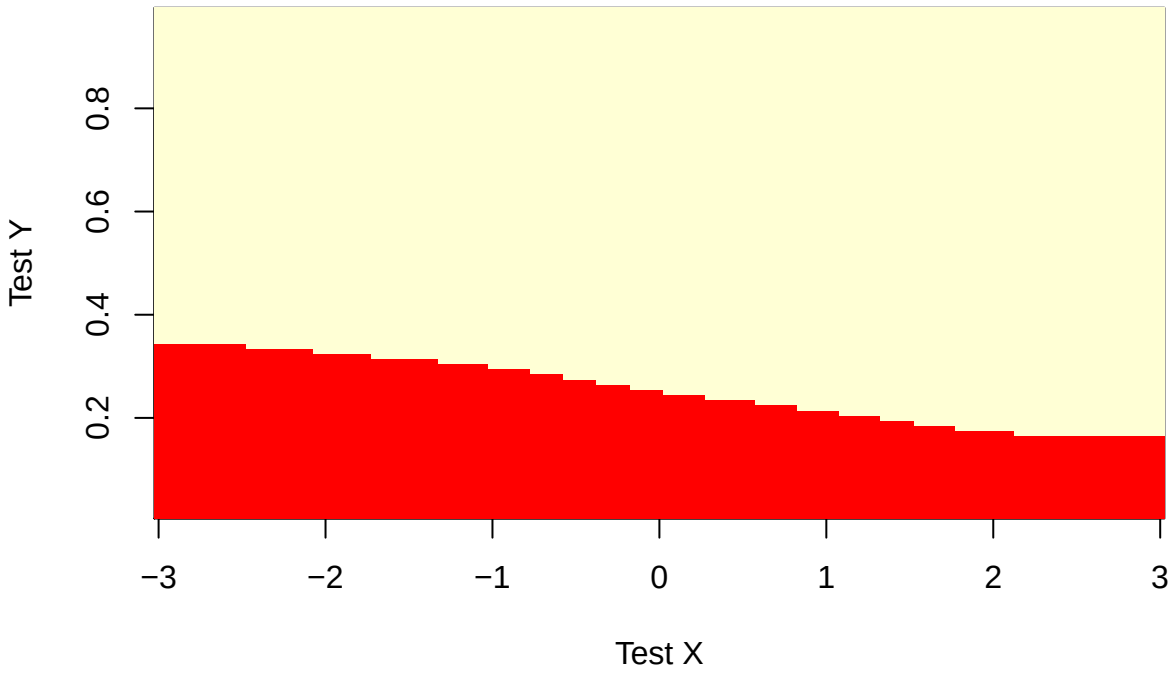
```r
# Define a loss matrix
Loss <- matrix(c(0,1,4,0),2,2, byrow=TRUE, dimnames=list(c("Treat","!Treat"),c("Sick","!Sick")))
Loss

# Decision rule matrix
Delta <- array(0,c(length(seq_x),length(seq_y)))

# This loop evaluates the probability that the patient is sick,
# given the test values x, y. The idea is to integrate out both
# pi and theta.
for(s in 1:length(seq_x)){ for(t in 1:length(seq_y)){

  x <- seq_x[s]
  y <- seq_y[t]

  # Evaluate probability
  pr <- gamma(x,y)

  # Save for probability table
  gamma_matrix[s,t] <- pr

  # Bayesian expected losses
  B0 <- pr*Loss["!Treat","Sick"] + (1-pr)*Loss["!Treat","!Sick"]
  B1 <- pr*Loss["Treat","Sick"]  + (1-pr)*Loss["Treat","!Sick"]

  # Optimal decision for this particular (x,y) combination.
  Delta[s,t] <- 1*(B1 < B0)

}}

# Equivalent threshold for probabilities.
thr <- (Loss[2,2] - Loss[1,2])/( (Loss[2,2] - Loss[1,2]) + (Loss[1,1] - Loss[2,1]))

# Color plotting of probabilities.
image(seq_x,seq_y,gamma_matrix, xlab= "Test X", ylab="Test Y")
contour(seq_x,seq_y,gamma_matrix, levels = thr,add=TRUE, lwd=2,col=4, labcex=1)
```




```
# Contour plot gives numerical values.
contour(seq_x,seq_y,gamma_matrix, xlab= "Test X", ylab="Test Y")
```




```
# Decision rule (only 1 or 0)
image(seq_x,seq_y,Delta, xlab= "Test X", ylab="Test Y")
```




I did some modifications to the original lab code and get the version that includes parameter $c$ and $d$. The modifications are all labeled with a comment `# Modified` in `R` script and `\\ Modified` in `stan` code.

I also modified some of the original R code, labeled by `# Modified`:

- Changed # of `Warmup` from T to B.
- Fixed a wrong label in drawing the traceplot of $\theta$.
- Changed the code for generating `pi0`. I'm not sure why the original code multiplied both parameters by 2. I just modified it to match our model.

Based on the traceplot of each parameter, we can see they all have a good mixing. Our decisions depend more on the result of Test Y than that of Test X since $Y$ is directly connected to $S$, while $X$ is indirectly connected to $S$ through parameters $\theta$ and $\pi$.