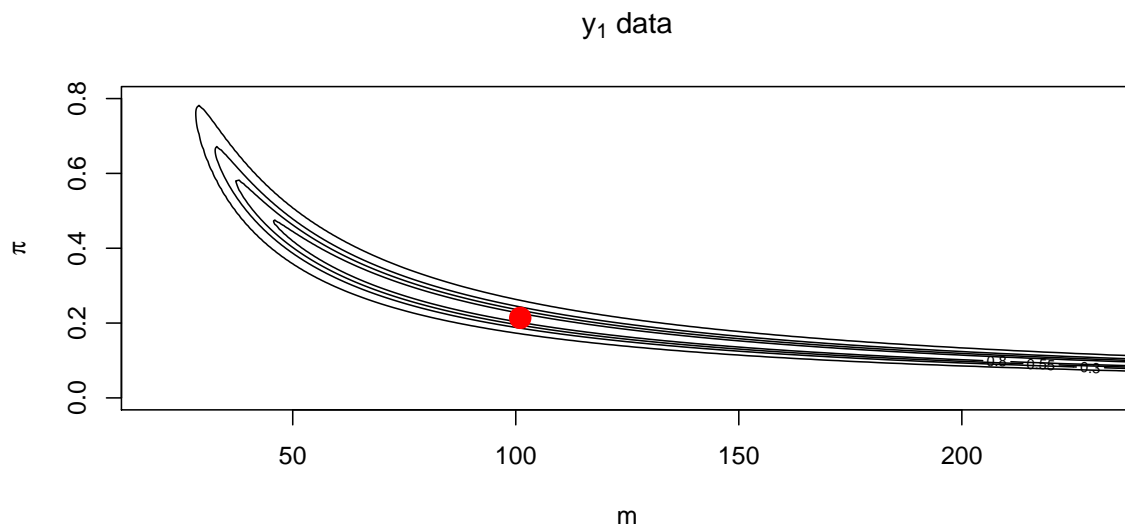


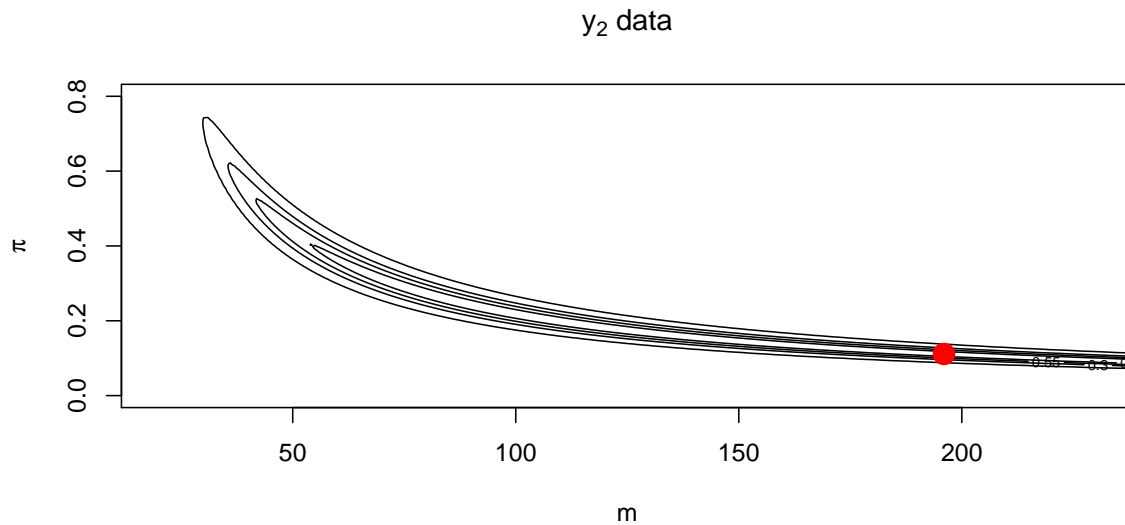
# Bayes Midterm 2

## Leslie Gains-Germain

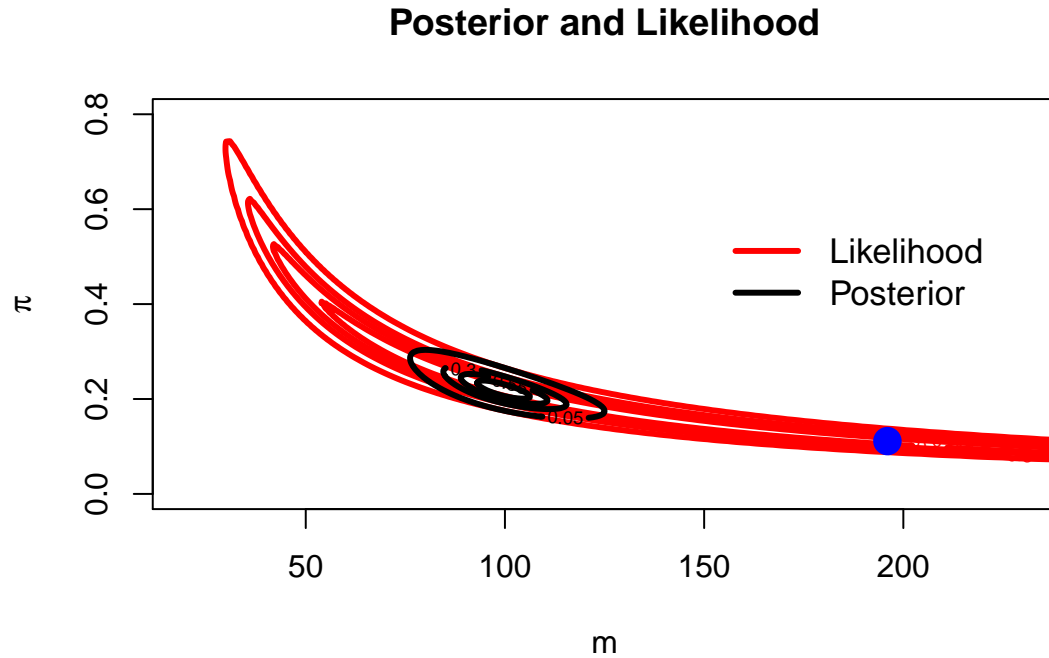
1. (a) The following example was inspired by Claire and Jordan's consulting project. Suppose there are five reservations in Montana. The number of children eligible to receive dental sealants on the reservation is unknown because it's hard to keep track of this population. A success is observed when a child comes to a dental clinic to have dental sealants applied. The number of children on each of the five reservations who had dental sealants applied in 2014 was observed. The probability of having the sealants applied and the number of eligible children is assumed to be the same for all five reservations.
- (b) The MLE occurs near  $m = 101, \pi = 0.21$ .



- (c) The MLE occurs near  $m = 196, \pi = 0.11$ .



- (d) i. The joint posterior distribution of  $(m, \pi)$  is shown below.



- ii. My work for deriving the complete conditionals for  $m$  and  $\pi$  is shown below.

$$\begin{aligned}
 p(\pi|y, m) &= \frac{p(\pi, y, m)}{p(y, m)} \\
 &\propto p(\pi, y, m) \\
 &\propto p(y|\pi, m)p(m, \pi) \\
 &\propto p(y|\pi, m)p(\pi) \\
 p(m|y, \pi) &= \frac{p(\pi, y, m)}{p(y, \pi)} \\
 &\propto p(\pi, y, m) \\
 &\propto p(y|\pi, m)p(m, \pi) \\
 &\propto p(y|\pi, m)p(m)
 \end{aligned}$$

- iii. My code for the Gibbs sampler is shown below.

```

#function to calculate complete conditional for pi on log scale
log.pi.cc.fun <- function(m.pi, y.vec){
  loglik.fun(m.pi, y.vec)
}

#check function
#log.pi.cc.fun(c(40, 0.5), y2.data)

#function to calculate complete conditional for m on log scale
log.m.cc.fun <- function(m.pi, y.vec){

```

```

    log.post.fun(m.pi, y.vec)
}

#check function
#log.m.cc.fun(c(40, 0.5), y2.data)

nchain <- 3
nsim <- 10000
m.pi.mat <- array(NA, dim=c(nsim, 2, nchain))

#keep track of acceptance ratios
jump.mat <- matrix(NA, nrow=nsim-1, ncol=2)

#specify starting values for each chain
m.pi.mat[1, 1:2, 1] <- c(100, 0.4)
m.pi.mat[1, 1:2, 2] <- c(50, 0.2)
m.pi.mat[1, 1:2, 3] <- c(150, 0.4)

#define standard deviations for normal proposal distributions
sd.scale <- c(10, 5)

set.seed(230923)

for (j in 1:nchain) {
  for (i in 2:nsim) {
    #set pi equal to the starting value
    pi <- m.pi.mat[i-1, 2, j]

    #now draw m from complete conditional with Metropolis Hastings Algorithm
    m.cur <- m.pi.mat[i-1, 1, j]
    m.cand <- rpois(1, lambda=m.cur)

    log.r.num.m <- log.m.cc.fun(c(m.cand, pi), y.vec = y2.data) +
      dpois(m.cur, lambda=m.cand, log = TRUE)

    log.r.denom.m <- log.m.cc.fun(c(m.cur, pi), y.vec = y2.data) +
      dpois(m.cand, lambda=m.cur, log = TRUE)

    log.r.m <- log.r.num.m - log.r.denom.m

    p.accept.m <- min(1, exp(log.r.m))

    u.vec <- runif(2)

    ifelse(u.vec[1] <= p.accept.m, m.pi.mat[i, 1, j] <- m.cand,
          m.pi.mat[i, 1, j] <- m.cur)

    jump.mat[i-1, 1] <- ifelse(u.vec[1] <= p.accept.m, 1, 0)

    #now fix m at its value in the ith iteration
    m <- m.pi.mat[i, 1, j]

    #draw pi from complete conditional with metropolis hastings algorithm

```

```

pi.cur <- m.pi.mat[i-1, 2, j]
pi.cand <- rbeta(1, 2, sd.scale[2])

log.r.num.pi <- log.pi.cc.fun(c(m, pi.cand), y.vec = y2.data) +
  dbeta(pi.cur, 2, sd.scale[2], log = TRUE)

log.r.denom.pi <- log.pi.cc.fun(c(m, pi.cur), y.vec = y2.data) +
  dbeta(pi.cand, 2, sd.scale[2], log = TRUE)

log.r.pi <- log.r.num.pi - log.r.denom.pi

p.accept.pi <- min(1, exp(log.r.pi))

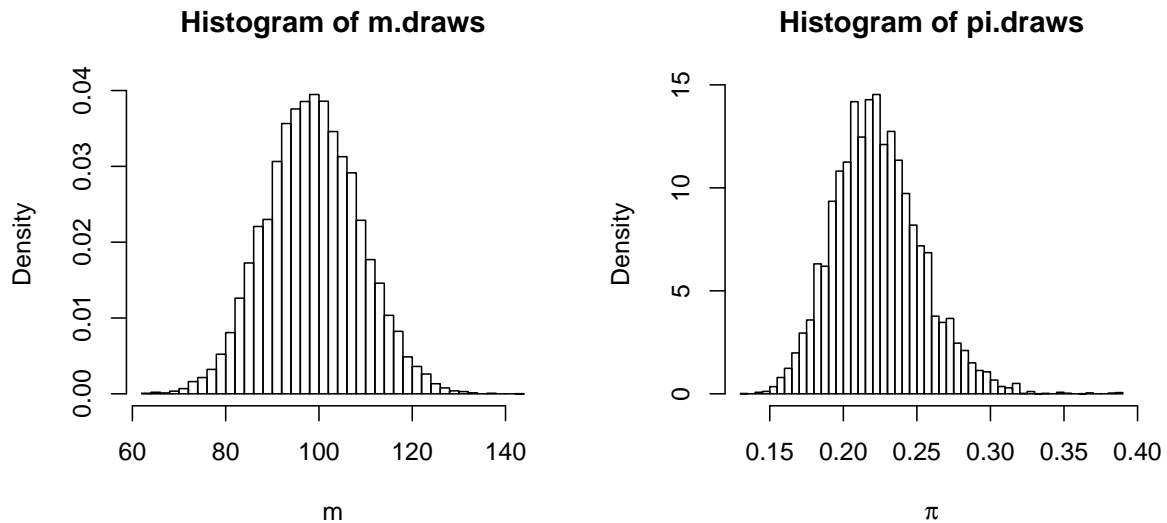
u.vec <- runif(2)

ifelse(u.vec[2] <= p.accept.pi, m.pi.mat[i, 2, j] <- pi.cand,
  m.pi.mat[i, 2, j] <- pi.cur)

jump.mat[i-1, 2] <- ifelse(u.vec[2] <= p.accept.pi, 1, 0)
}
}

```

- (e) The marginal posterior distributions for  $m$  and  $\pi$  are shown below. The posterior probability that  $m > 100$  is 0.443 and the posterior probability that  $\pi < 0.3$  is 0.989. Since  $m$  and  $\pi$  are independent, the joint posterior probability that  $m > 100$  and  $\pi < 0.3$  is  $0.443 * 0.989 = 0.438$ .



- (f) disads
- (g) The hierarchical version of the model puts priors on  $\lambda$  and priors on parameters of a  $Beta(\alpha, \beta)$  distribution. I chose to put priors on the prior mean,  $\eta = \frac{\alpha}{\alpha + \beta}$  and the somewhat approximate prior standard deviation,  $\sigma = \sqrt{\frac{1}{\alpha + \beta}}$ , kind of like what we did in

the Beta-Binomial example with the stomach cancer data.

$$\begin{aligned}y_i &\sim \text{Bin}(m, \pi) \\ \pi &\sim \text{Beta}\left(\frac{\eta}{\sigma^2}, \frac{1-\eta}{\sigma^2}\right) \\ \sigma &\sim \text{Uniform}(0, 100) \\ \eta &\sim \text{Uniform}(0, 1) \\ \lambda &\sim \text{Uniform}(0, 500)\end{aligned}$$

(h) The JAGs code I used to obtain draws from the above model is shown below.

```
##write model file first
```

```
cat("
model
{
  for(i in 1:N)
  {
    y[i] ~ dbin(pi, m)
  }

  pi ~ dbeta(eta/sigma^2, (1-eta)/sigma^2)
  eta ~ dunif(0, 1)
  sigma ~ dunif(0, 100)

  m ~ dpois(lambda)
  lambda ~ dunif(0, 500)
}",
file="model1.jags")
```

```
##jags call
```

```
library(R2jags)
set.seed(52)
```

```
dental.data <- list(N=length(y2.data), y=y2.data)
```

```
inits <- list(list(pi=0.4, eta = .00003, m = 100, lambda = 500,
                  sigma = 1.20),
              list(pi=0.2, eta = .00003, m = 50, lambda = 150,
                  sigma = 2.24),
              list(pi=0.4, eta = .00003, m = 150, lambda = 300,
                  sigma = 1.58))
```

```
n.chain <- 3
```

```
#warmup
```

```
warmup.model1 <- jags.model("model1.jags", data=dental.data, n.chains=n.chain, inits= inits, n.adapt=
```

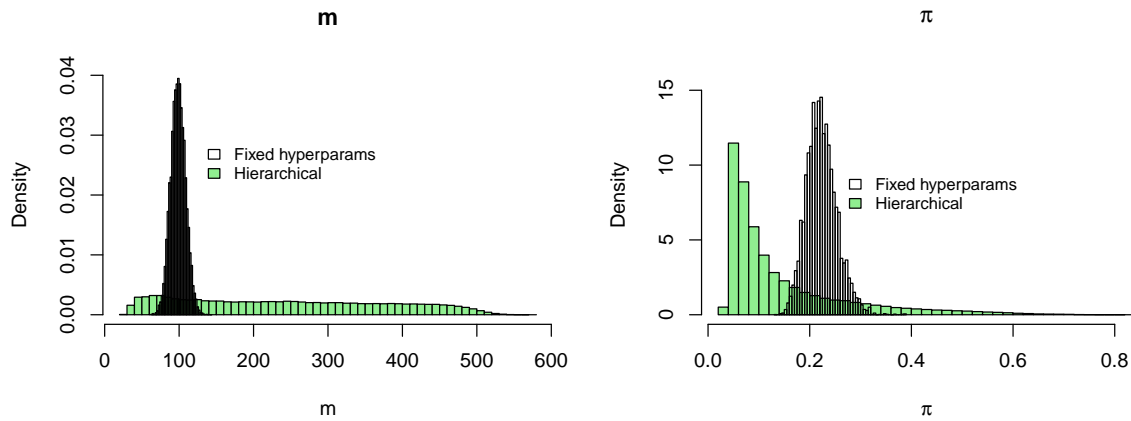
```
#parameters to save
```

```
params <- c("pi", "eta", "sigma", "m", "lambda")
```

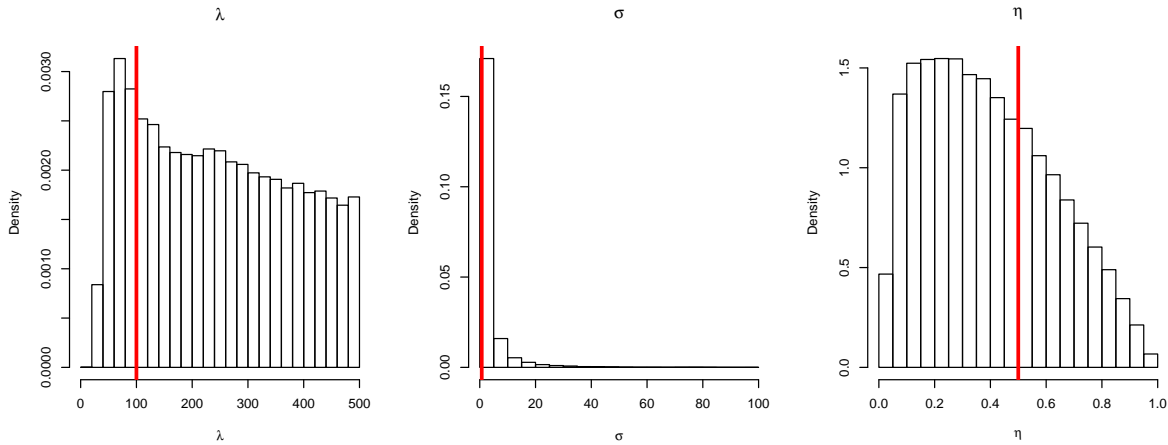
```
n.iter=50000
```

```
#running the model for real
model1 <- coda.samples(warmup.model1, params, n.iter=n.iter)
```

- (i) The plots below compare the posterior distributions for  $\pi$  and  $m$  obtained in part (f) to those from the hierarchical model. There is much more variability in the posterior draws for  $m$  and  $\pi$  in the hierarchical model compared to the model with fixed hyperparameters. I think this makes sense because when the hyperparameter values are approximated, the model does not incorporate uncertainty in these parameters. With such a small sample size, the additional source of uncertainty accounted for in the hierarchical model makes a big difference in the results.



The posterior distributions of the hyperparameters are shown below, with the fixed values used in (e).



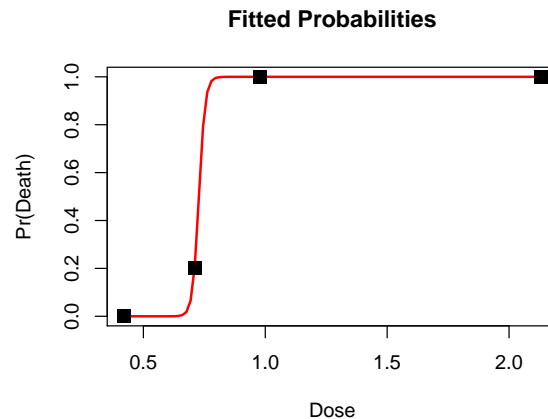
(j)

2. (a) The simple logistic regression model for estimating the dose-death relationship is as follows. I treat dose as a continuous predictor variable. The empirical probabilities of death at each dose as well as the fitted probabilities from the `glm()` model are shown on the plot below. The dose at which half of the frogs are estimated to die is  $\frac{0.5+56.97}{78.29} = 0.734$

grams per ml.

$$y_{ij} \sim \text{Bin}(5, \pi_j)$$
$$\text{logit}(\pi_j) = \alpha + \beta x_j$$

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```



(b) The model I will use to do the analysis in a Bayesian framework is shown below.

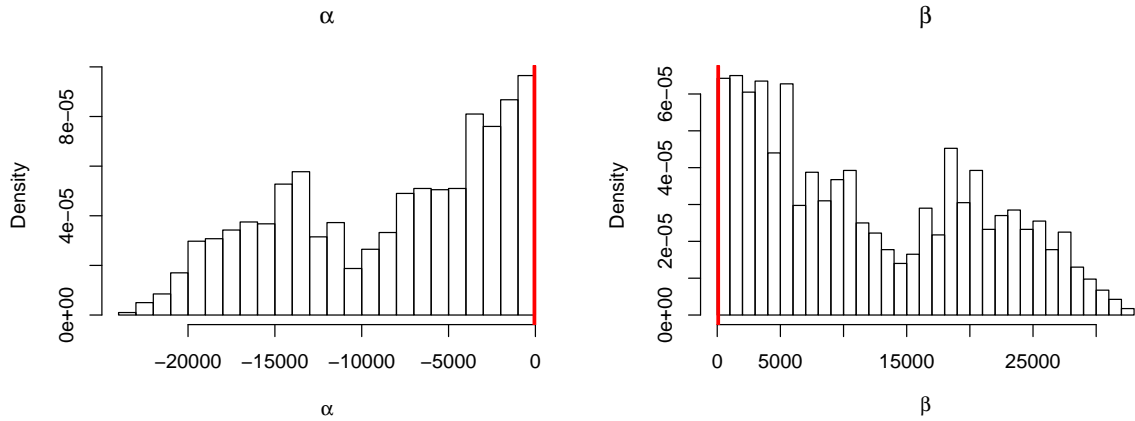
$$y_{ij} \sim \text{Bern}(\text{logit}^{-1}(\alpha + \beta x_j))$$
$$\alpha \propto 1$$
$$\beta \propto 1$$

(c) The model code is shown below.

```
data {  
  int<lower=0> N;  
  vector[N] x;  
  int<lower=0,upper=1> y[N];  
}  
parameters {  
  real alpha;  
  real beta;  
}  
model {  
  y ~ bernoulli_logit(alpha + beta * x);  
}
```

```
require(rstan)  
set.seed(23)  
data <- with(frog.data, list(y = death, x = dose, N = length(death)))  
  
model2 <- stan_model(file = "~/Documents/Stat532/exams/exam2/model2.stan",  
                      model_name = "model2")  
samp2 <- sampling(model2, chains = 4, iter = 2000, data = data)
```

- (d) The posterior draws for the y-intercept ( $\alpha$ ) and the dose effect ( $\beta$ ) are shown below. The red line indicates the estimate from the above generalized linear model.



- (e) Discuss and explain
- (f) The plots of the posterior draws for  $\alpha$  and  $\beta$  are shown below for the model with standardized dose as the predictor variable. The posterior draws for  $\alpha$  have changed, but this is expected because  $\alpha$  is now the log odds of death at the average dose. Standardizing dose does not appear to change inference about the effect of dose on the log odds of survival. Also, the efficiency of the sampler has not changed noticeably.

