Applied Statistics (BUGS)

Yasuhiro Omori

A1A2 term, 2019-2020

University of Tokyo

BUGS

4.2 Normal distribution

```
# First we specify the statistical model
model
{
# time_i ~ N(mu, sigma^2),  i = 1, ..., n
for(i in 1:n)
{
time[i] ~ dnorm(mu, inv.sigma2)
}
# Prior:  mu ~ N(0, 10^6)
mu ~ dnorm(0, 0.000001)
# Prior:  sigma^-2 ~ G(10^-4, 10^-4)
inv.sigma2 ~ dgamma(0.0001, 0.0001)
sigma2 <- 1/inv.sigma2
```

```
sigma <- sqrt(sigma2)
# the end of the model statement
}
# set initial values for MCMC
list( mu = 0, inv.sigma2 = 1)
# define variables for the dataset
list(n = 20)
# define variable time
time[]
182
...
365
END
```

2.4 Binomial distribution

```
# First we specify the statistical model
model
{
# y ~ Binomial(n, p)
y ~ dbin(p, n)
# Prior:  p ~ Beta(a, b)
p ~ dbeta(a,b)
# pr.ge.half=I(p > 0.5) to estimate Pr(p > 0.5|y)
pr.ge.half <- step(p-0.5)
# Generate y.pred ~ Binomial(20, p) where p ~ π(p|y)
y.pred ~ dbin(p, 20)
# the end of the model statement
}
```

```
# define variables for the dataset
list(n = 27, y = 11, a = 3.26, b = 7.18)
# set initial values for MCMC
list( p= 0.4, y.pred=5 )
```

3.3 Poisson distribution

```
# First we specify the statistical model
model
{
# y ~ Poisson(λ_y),  λ_y = eo × λ
y ~ dpois(lambda.y)
lambda.y <- eo*lambda
# Prior:  λ ~ G(α, β)
lambda ~ dgamma(alpha, beta)
# the end of the model statement
}
# define variables for the dataset
list( y=1, eo=66, alpha=16, beta=15174)
# set initial values for MCMC
list( lambda = 0.001)
```

6.9 Cauchy Distribution

```
# First we specify the statistical model
model
{
for(i in 1:n)
{
# Likelihood:  lik_i = {1 + (diff_i - mu)^2 / sigma^2}^(-1)
lik[i] <-
1/(sigma*(1+((diff[i]-mu)/sigma)*((diff[i]-mu)/sigma)))
# 0's trick.  Define lambda_i = -log(lik_i) + 10000 > 0
lam[i] <- -log(lik[i])+10000
# Define an auxiliary variable zeros_i ≡ 0
zeros[i] <- 0
```

Likelihood: $lik_i = \left\{1 + \frac{(diff_i - \mu)^2}{\sigma^2}\right\}^{-1}$

0's trick. Define $\lambda_i = -\log(lik_i) + 10000 > 0$

Define an auxiliary variable $zeros_i \equiv 0$

```
# zeros_i ~ Poisson(λ_i),  Pr(zeros_i = 0) = lik_i * exp(−10000)
zeros[i] ~ dpois(lam[i])
# Prior:   μ ~ N(0, 10^6)
mu ~ dnorm(0, 0.000001)
# Prior:   σ ~ G(10^-3, 10^-3)
sigma ~ dgamma(0.001, 0.001)
lsigma <- log(sigma)
# the end of the model statement
}
# set initial values for MCMC
list( mu = 0, sigma=1 )
# define variables for the dataset
list(n = 15)
```

```
# define variable diff
diff[]
-67
...
75
END
```

6.10 Heart Transplant Data

```
# First we specify the statistical model
model
{
for(i in 1:n)
{
# Likelihood when state_i = 0, transplant_i = 0
# log(lik_i) = log p + p log λ − (p + 1) log(λ + survtime_i)
llk_nt_0[i] <-
log(p)+p*log(lambda)-(p+1)*log(lambda+survtime[i])
# Likelihood when state_i = 1, transplant_i = 0
# log(lik_i) = p log λ − p log(λ + survtime_i)
llk_nt_1[i] <-
p*log(lambda)-p*log(lambda+survtime[i])
```

```
# Likelihood when state_i = 1, transplant_i = 0
# log(lik_i) = log p + log τ + p log λ
#    −(p + 1) log(λ + timetotransplant_i + τ × survtime_i)
llk_t_0[i] <-
log(p)+log(tau)+p*log(lambda)
-(p+1)*log(lambda+timetotransplant[i]+tau*survtime[i])
# Likelihood when state_i = 1, transplant_i = 1
# log(lik_i) = p log λ
#    −p log(λ + timetotransplant_i + τ × survtime_i)
llk_t_1[i] <- p*log(lambda)
-p*log(lambda+timetotransplant[i]+tau*survtime[i])
```

```
# Likelihood:
loglk[i] <-
   (1-transplant[i])*(1-state[i])*llk_nt_0[i]
  +(1-transplant[i])* state[i] *llk_nt_1[i]
  + transplant[i] *(1-state[i])*llk_t_0[i]
  + transplant[i] * state[i] *llk_t_1[i]
# 0's trick.  Define mu_i = -log(lk_i) + 10000 > 0
mu[i] <- -loglk[i]+10000
# Define an auxiliary variable zeros_i ≡ 0
zeros[i] <- 0
# zeros_i ~ Poisson(mu_i),  Pr(zeros_i = 0) = lk_i * exp(-10000)
zeros[i] ~ dpois(mu[i])
# the end of for loop
}
```

```
# τ = exp(log τ),  λ = exp(log λ),  p = exp(log p)
tau <- exp(ltau)
lambda <- exp(llambda)
p <- exp(lp)
# Prior:  log τ ~ N(0, 10^6)
ltau ~ dnorm(0, 0.000001)
# Prior:  log λ ~ N(0, 10^6)
llambda ~ dnorm(0, 0.000001)
# Prior:  log p ~ N(0, 10^6)
lp ~ dnorm(0, 0.000001)
```

```
# Estimate Pr(τ ≤ 1) using tau.le.1=I(τ ≤ 1)
tau.le.1 <- step(1 - tau)
# S(t):  survival function.  t = 1,...,240
for(t in 1:240)
{
S[t] <- exp( p*log(lambda) - p*log(lambda + t) )
time[t] <- t
}
# the end of the model statement
}
# set initial values for MCMC
list( ltau = 0, llambda=0, lp=0 )
# define variables for the dataset
list(n = 82)
```

```
# define variables:  survtime, transplant,
timetotransplant, state
survtime[] transplant[] timetotransplant[] state[]
49 0 0 0
...
43 1 5 1
END
```

7 Hierarchical modeling

```
# First we specify the statistical model
model
{
for(i in 1:n)
{
# y_i|λ ~ POI(e_i × λ)
y[i] ~ dpois(e.lambda[i])
e.lambda[i] <- e[i]*lambda
} # Prior:  λ ~ Gamma(0.001, 0.001)
lambda ~ dgamma(0.001, 0.001)
```

Section 7.4 Equal mortality rates

```
# Posterior predictive analysis
# Pr(Y_{i,pred} < Y_i|data) < 0.05?
# or Pr(Y_{i,pred} > Y_i|data) < 0.05?
# If yes, the model may not be valid for Y_i.
for(i in 1:n)
{
ypred[i] ~ dpois(e.lambda[i])
pright[i] <- step(ypred[i] - y[i])
pleft[i] <- step(y[i] - ypred[i])
}
# Initial values.
list( lambda = 1 )
# Initialize other values using "gen inits".
```

```
# Data
list(n=94)
# Data
e[] y[]
532 0
...  ...
12131 17
END
```

Section 7.5 Modeling a prior belief of exchangeabiliy

```
# First we specify the statistical model
model
{
for(i in 1:n)
{
# y_i|λ_i ~ POI(e_i × λ_i)
# λ_i ~ Gamma(α, α/μ),  E(λ_i) = μ,  Var(λ_i) = μ^2/α
y[i] ~ dpois(e.lambda[i])
e.lambda[i] <- e[i]*lambda[i]
lambda[i] ~ dgamma(alpha, beta)
loge[i] <- log(e[i])
}
beta <- alpha*mu.inv
```

```
# Prior:  μ ~ IG(0.001, 0.001)
mu.inv ~ dgamma(0.001, 0.001)
mu <- 1/mu.inv
# Prior:  α ~ g(α) = z0/(α + z0)^2,  z0 = 0.53.
# z0 is the median of α.
alpha <- x-z0
z0 <- 0.53
# f(x) = z0/x^2, x > z0, (x = α + z0).
x ~ dpar(1, z0)
lalpha <- log(alpha)
lmu <- log(mu)
```

```
# Compute shrinkage size B_i toward the pooled
# estimate μ:   E(λ_i|y,α,μ) = (1 − B_i) y_i/e_i + B_i μ
# where B_i = α/(α + e_i μ)
for(i in 1:n)
{
B[i] <- alpha/(alpha+e[i]*mu)
}
# Posterior predictive analysis
for(i in 1:n)
{
ypred[i] ~ dpois(e.lambda[i])
pright[i] <- step(ypred[i] - y[i])
pleft[i] <- step(y[i] - ypred[i])
}
```

Section 7.5 Modeling a prior belief of exchangeabiliy

```
# Initial values.
list( mu.inv=1, x=1 )
# Initialize other values using "gen inits".      #
Data
list(n=94)
# Data
e[] y[]
532 0
...  ...
12131 17
END
```

9.2 Regression model

```
# First we specify the statistical model
model
{
for(i in 1:n)
{
# log time_i ~ N(mu, sigma^2)
# where mu_i = beta_1 + beta_2 x nesting_i + beta_3 x size_i + beta_4 x status_i
logtime[i] <- log(time[i])
logtime[i] ~ dnorm(mu[i], inv.sigma2))
mu[i] <- b[1] + b[2]*nesting[i] + b[3]*size[i] +
b[4]*status[i]
}
```

```
# Prior:  β_i ~ N(0, 10^6),  i = 1, 2, 3, 4
b[1] ~ dnorm(0, 0.000001)
b[2] ~ dnorm(0, 0.000001)
b[3] ~ dnorm(0, 0.000001)
b[4] ~ dnorm(0, 0.000001)
# Prior:  σ^-2 ~~ IG(10^-4, 10^-4)
inv.sigma2 ~ dgamma(0.0001, 0.0001)
sigma2 <- 1/inv.sigma2
sigma <- sqrt(sigma2)
```

```
# Prediction of the new mean μ_pi and
# the new observation y_pi ~ N(μ_pi, σ^2),  i = 1,2,3,4
for(i in 1:4)
{
mu_p[i] <-b[1]+b[2]*nesting_p[i]+b[3]*size_p[i]
+b[4]*status_p[i]
y_p[i] ~ dnorm(mu_p[i], inv.sigma2)
}
# Prediction of the new observation y*_i ~ N(μ_i, σ^2)
for(i in 1:n)
{
ystar[i] ~ dnorm(mu[i], inv.sigma2)
# Compute pright_i = I(ystar_i > log time_i)
# to estimate Pr(ystar_i > log time_i)
```

```
pright[i] <- step(ystar[i] - logtime[i])
# Compute pleft_i = I(ystar_i < log time_i)
# to estimate Pr(ystar_i < log time_i)
pleft[i] <- step(logtime[i] - ystar[i])
}
# the end of the model statement
}
# define variables for the dataset
list(n = 62)
```

```
# define variables:
time[] nesting[] size[] status[]
3.030 1.000 0 1
...
1.000 1.000 1 1
END
nesting_p[] size_p[] status_p[]
4 0 0
...
4 1 1
END
# set initial values for MCMC
list( b=c(0, 0, 0, 0), inv.sigma2 = 1, y_p=c( 0,
0, 0, 0))
```

11.4 Change point model

Section 11.4 Change point model

```
# First we specify the statistical model
model
# D_t ~ POI(mu_t),  mu_t = exp(beta_1 + I(t >= t_0) * beta_2),
# t_0 : change point
# mu_t = exp(beta_1) for t < t_0,  mu_t = exp(beta_1 + beta_2) for t >= t_0.
{
for( year in 1 :  N )
{
D[year] ~ dpois(mu[year])
log(mu[year]) <- b[1] + step(year - changeyear) * b[2]
}
```

```
# Prior:  β1 ~ N(0,10^6), β2 ~ N(0,10^6)
b[1] ~ dnorm( 0.0,1.0E-6)
b[2] ~ dnorm( 0.0,1.0E-6)
# Prior:  Pr(t0 = k) = 1/N, k = 1,...,N.
for( year in 1 :  N )  p[year] <- 1/N )
changeyear ~ dcat(p[])
# Initial values
list(b = c(0, 0), changeyear = 50)
# Data
list( N = 112, D=c(4,5,4,1,0,4,3,4,0,6,...,0,0))
```