

# Introduction to Bayesian Modeling in R

Will Barnett



Neptune and Company, Inc.  
[www.neptuneinc.org](http://www.neptuneinc.org)



# Presentation Outline

- What is Bayesian Modeling?
- Example Using R and JAGS
- Example Using MCMCpack
- Where to Learn More



# What does Bayesian mean?

- “The conditional distribution of parameters and unobserved data, given observed data...” (Gelman, 2008)

$$\begin{array}{c} \text{Posterior} \\ \boxed{P(\textit{parms} \mid \textit{data})} \end{array} = \frac{\begin{array}{c} \text{Likelihood} \\ \boxed{P(\textit{data} \mid \textit{parms})} \end{array} * \begin{array}{c} \text{Prior} \\ \boxed{P(\textit{parms})} \end{array}}{P(\textit{data})}$$



# Basic Terminology

- Prior: the distribution of any unknown parameters before any data is known
- Likelihood: A model of how your data might behave for a given set of parameters
- Posterior: The distribution of parameters given the prior information and the data



# Why should we care?

- Lots of literature out there. A few advantages (from a biased observer):
  - Easily incorporate prior/expert knowledge
  - Can fit complex models that are almost impossible with classical methods
  - Much simpler to interpret results



# Quick Example

- Flip a quarter 100 times, get heads 60 times. Is the coin fair?
  - The 95% confidence interval for  $\hat{p}$  is (0.504, 0.696).
- What if you are sure it's a real quarter? Do you really want to flip it another 100 times?
  - Bayesian statistics can quantify the prior information about the fairness of the coin.



# Example with R / JAGS

- Dataset from Royle / Dorazio, 2006
  - Moth mortality based on dose of poison, sex (M/F)
  - Typical logistic regression model
  - Use Markov Chain Monte Carlo (MCMC) to arrive at posterior distribution, with JAGS/BUGS

# Example with R / JAGS

- BUGS/JAGS are almost identical open-source platforms for specifying and running Bayesian models
  - Both are stand-alone programs that must be installed
- The packages 'R2WINBUGS' and 'rjags' allow R users to access either from within R
- Very flexible language





# Example with R / JAGS, cont'd

**Moth Data (20 per sex/dose combination):**

Mortality	Sex	Dose
1	male	1
4	male	2
9	male	4
13	male	8
18	male	16
20	male	32
0	female	1
2	female	2
6	female	4
10	female	8
12	female	16
16	female	32



# Example with R / JAGS, cont'd

**R code:**

```
nSamp = 20
```

```
dataList <- list(n = nrow(mothData),  
                 nSamp = nSamp, y = mothData$y,  
                 x = mothData$ldose,  
                 z = mothData$sex)
```

```
params <- list('beta0','beta1','beta2')
```

```
inits <- list(beta0 = rnorm(1), beta1 = rnorm(1),  
              beta2 = rnorm(1))
```



# Example with R / JAGS, cont'd

**JAGS/BUGS code, saved in .txt / .jag / .bug file:**

```
model {  
  beta0 ~ dnorm(0, 0.001)  
  beta1 ~ dnorm(0, 0.001)  
  beta2 ~ dnorm(0, 0.001)  
  for (i in 1:n) {  
    y[i] ~ dbin(p[i], nSamp)  
    logit(p[i]) <- beta0 + beta1 * x[i] + beta2 * z[i]  
  }  
}
```



# Example with R / JAGS, cont'd

**R code:**

```
params <- c("beta0","beta1","beta2","p")  
library(rjags); # call 'coda' package  
jags.mod <- jags.model(file = "model.jag", data =  
                      dataList, inits = inits, n.chains = 3)  
jags.samps <- coda.samples(model = jags.mod,  
                          variable.names = params, n.iter =  
                          5000, thin = 10, n.burnin = 1000)
```



# Example with R / JAGS, cont'd

R code:

```
summary(jags.samps)
```

```
> summary(jags.samps)
```

```
Iterations = 1010:6000
```

```
Thinning interval = 10
```

```
Number of chains = 3
```

```
Sample size per chain = 500
```

1. Empirical mean and standard deviation for each variable,  
plus standard error of the mean:

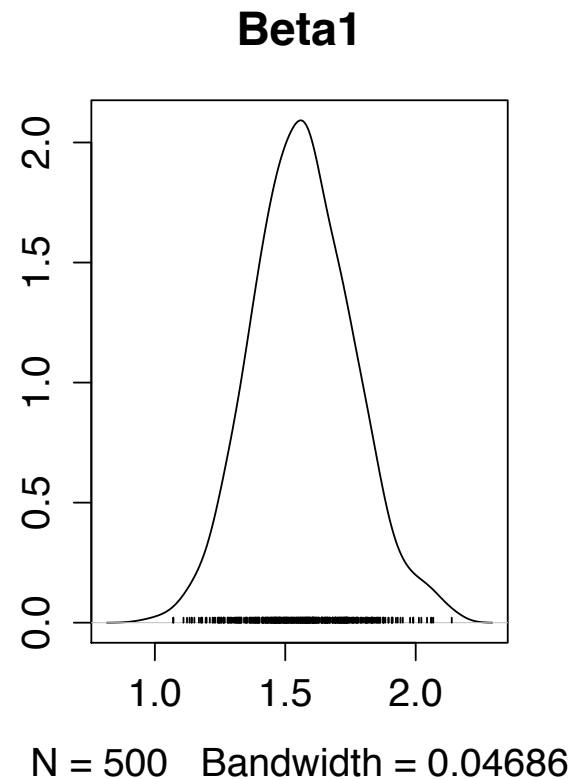
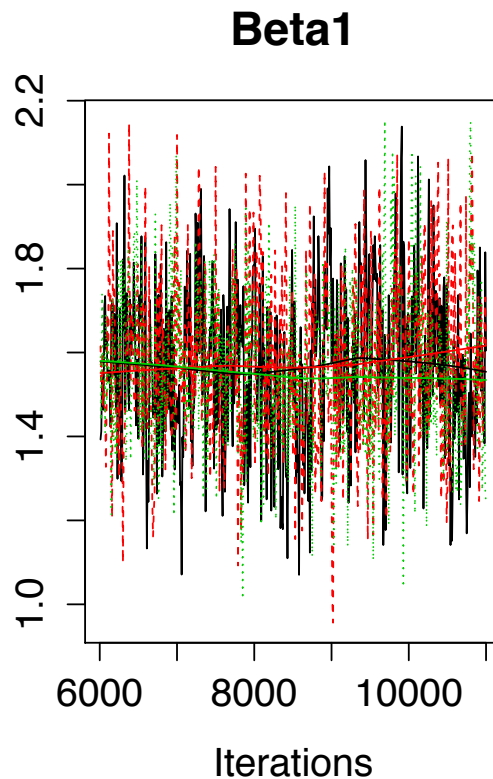
	Mean	SD	Naive SE	Time-series SE
beta0	-4.74784	0.79679	0.0205731	0.0424130
beta1	1.57506	0.19889	0.0051353	0.0081902
beta2	1.16266	0.36824	0.0095080	0.0197403
p[1]	0.08620	0.03038	0.0007845	0.0009303
p[2]	0.21310	0.04824	0.0012456	0.0013715



# Example with R / JAGS, cont'd

R code:

```
plot(jags.samps[,2], main = "Beta1")
```



# Example with MCMCpack

- MCMCpack is an R package that allows fitting of Bayesian models natively (without calling external program)
- Has several built-in functions for regression modeling
- You can also write your own prior and posterior, and it does the grunt MCMC work with Metropolis-Hastings algorithm
  - If posterior isn't closed-form, this can be a mess



# Example with MCMCpack, cont'd

**R code:**

```
library(MCMCpack)
```

```
...
```

```
# re-working data frame to include 1's and 0's
```

```
...
```

```
mcmcpack.samps <- MCMClogit(y ~ Idose +  
                             sex, data = mothDataAdj,  
                             burnin = 1000, mcmc = 5000,  
                             thin = 10, b0=0, B0=.001)
```





# Example with MCMCpack, cont'd

R code:

```
summary(mcmcpack.samps) # a 'mcmc' object
```

```
> summary(mcmcpack.samps)
```

```
Iterations = 1001:5991
```

```
Thinning interval = 10
```

```
Number of chains = 1
```

```
Sample size per chain = 500
```

1. Empirical mean and standard deviation for each variable,  
plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
(Intercept)	-3.541	0.5042	0.022547	0.02550
ldose	1.570	0.1967	0.008799	0.01027
sexmale	1.102	0.3762	0.016823	0.01934



# Example with GLM fit

R code:

```
glm.fit <- glm(y / total ~ ldose + sex, data =  
              mothData, family = "quasibinomial")  
## An overdispersed binomial actually fits this  
better than the typical binomial.  
summary(glm.fit)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	-3.4732	0.3597	-9.655	4.79e-06	***
ldose	1.5353	0.1452	10.574	2.24e-06	***
sexmale	1.1007	0.2732	4.029	0.00298	**

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1



# Comparisons of Models

Coefficient	JAGS/BUGS	MCMCpack	GLM
Beta0	-4.66 (0.733)*	-3.54 (0.504)	-3.47 (0.360)
Beta1 (log-dose)	1.57 (0.191)	1.57 (0.200)	1.54 (0.145)
Beta2 (sex)	1.11 (0.354)	1.10 (0.376)	1.10 (0.273)

\*The JAGS/BUGS intercept seems off. There is an available R script from Neptune that shows some diagnostics for ‘mcmc’ output in R.



# Brief Other Example

- Andrew Finley (Michigan State) has a neat example about forest biomass estimation
- [http://blue.for.msu.edu/CSTAT\\_13/](http://blue.for.msu.edu/CSTAT_13/)
  - MCMC Short course notes / slides / code
- Forest inventory data over a decade, with LANDSAT covariates
- Fits non-spatial regression model with `lm()`, non-spatial regression model in JAGS, spatial regression model with exponential covariance structure and random effects



# Brief Other Example, cont'd

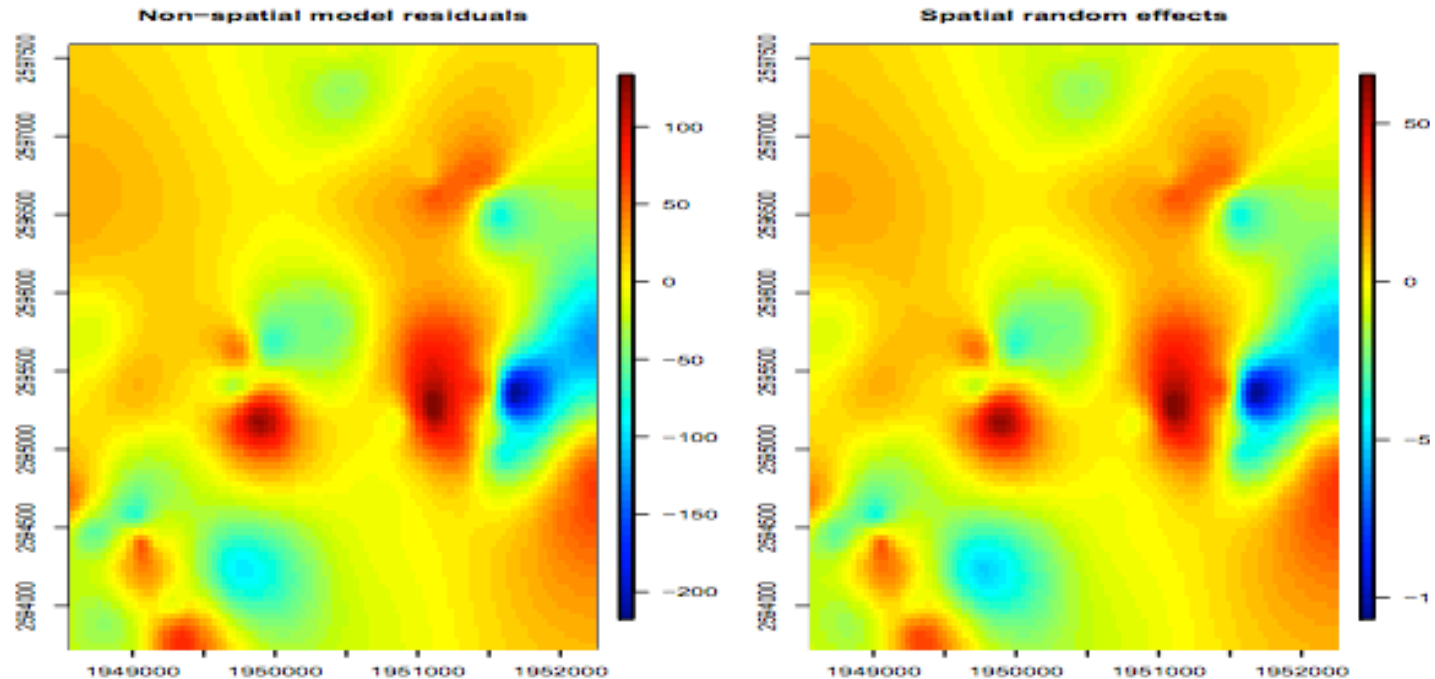


Figure 7: Interpolated surfaces of non-spatial model residuals (left) and spatial random effects posterior mean (right) at each location.

# Brief Other Example, cont'd

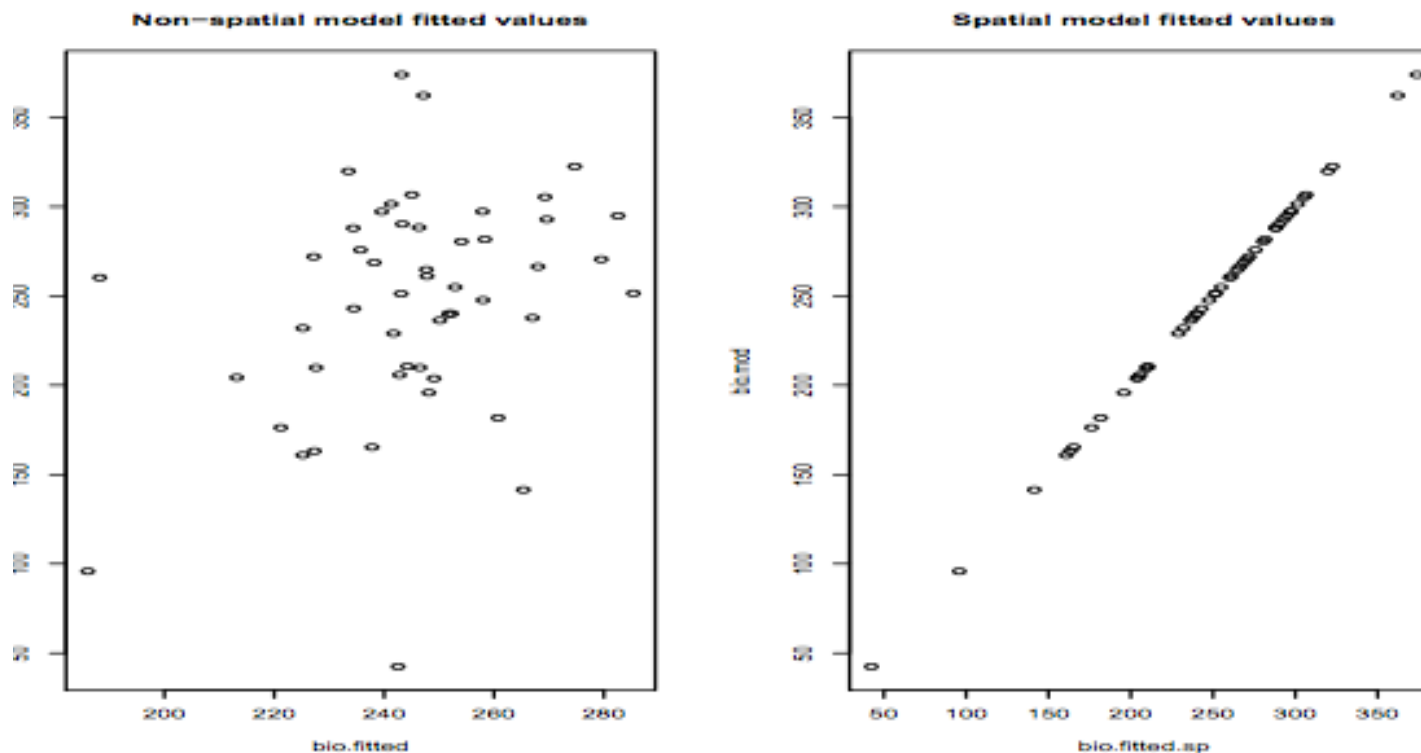


Figure 8: (left) Non-spatial model fitted values (right) spatial model fitted values.

# Resources

- Gelman et al, Bayesian Data Analysis
- Gelman and Hill, Data Analysis Using Regression and Multilevel/Hierarchical Models
- Royle and Dorazio, Hierarchical Modeling and Inference in Ecology
  - <http://www.mbr-pwrc.usgs.gov/pubanalysis/roylebook/index.html>
- OpenBUGS Examples:
  - <http://www.openbugs.net/w/Examples>
- And of course, CRAN Bayesian task view:
  - <http://cran.r-project.org/web/views/Bayesian.html>

