

# ADVANCED BAYESIAN MODELING

# Ebola Data Example: Overdispersed Model

# Ebola Outbreaks Data

$y_i$  = deaths in outbreak  $i$ , out of  $n_i$  human cases

$x_{i1}, x_{i2}, x_{i3}$  = indicators of ebolavirus type (BDBV, EBOV, SUDV)

$x_{i4}$  = year outbreak began (centered and scaled)

$$i = 1, \dots, 17$$

## Model Allowing Overdispersion

$$y_i \mid \beta, X_i \sim \text{indep. Bin}(n_i, p_i)$$

$$\text{logit}(p_i) = X_i \beta + \epsilon_i \quad \epsilon_i \sim \text{iid N}(0, \sigma_\epsilon^2)$$

Random effects  $\epsilon_i$  allow for overdispersion.

We will use the same scaled- $t_1$  priors on coefficients in  $\beta$  as before, and a flat hyperprior on  $\sigma_\epsilon$  (approximated by a uniform).

In ebola2.bug:

```
model {  
  for (i in 1:length(deaths)) {  
    deaths[i] ~ dbin(prob[i], cases[i])  
    logit(prob[i]) <- betavirus[virus[i]] + betayear*yearscaled[i] + epsilon[i]  
  
    epsilon[i] ~ dnorm(0,1/sigmaepsilon^2)  
  
    deathsrep[i] ~ dbin(prob[i], cases[i])  
  }  
  
  for (j in 1:max(virus)) {  
    betavirus[j] ~ dt(0, 0.01, 1)  
  }  
  betayear ~ dt(0, 0.16, 1)  
  
  sigmaepsilon ~ dunif(0,10)  
}
```

Data is set up as before, but initializations of the 4 chains now include `sigmaepsilon`:

```
> d2 <- list(deaths = ebola$Deaths,  
+           cases = ebola$Cases,  
+           virus = unclass(ebola$Virus),  
+           yearscaled = as.vector(scale(ebola$Year, scale=2*sd(ebola$Year))))  
  
> inits2 <- list(list(betavirus=c(10,10,10), betayear=10, sigmaepsilon=0.01),  
+               list(betavirus=c(10,10,-10), betayear=-10, sigmaepsilon=9),  
+               list(betavirus=c(10,-10,10), betayear=-10, sigmaepsilon=0.01),  
+               list(betavirus=c(10,-10,-10), betayear=10, sigmaepsilon=9))
```

```

> m2 <- jags.model("ebola2.bug", d2, inits2, n.chains=4, n.adapt=1000)
Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 17
  Unobserved stochastic nodes: 39
  Total graph size: 221

Initializing model

|+++++| 100%

> update(m2, 1000) # burn-in
|*****| 100%

> x2 <- coda.samples(m2, c("betavirus", "betayear", "sigmaepsilon"), n.iter=2000)
|*****| 100%

```

```
> gelman.diag(x2, autoburnin=FALSE)
```

Potential scale reduction factors:

	Point est.	Upper C.I.
betavirus[1]	1.05	1.11
betavirus[2]	1.02	1.04
betavirus[3]	1.01	1.02
betayear	1.05	1.14
sigmaepsilon	1.01	1.01

Multivariate psrf

1.05

Close to convergence, but continue burn-in just to be sure ...



```
> x2 <- coda.samples(m2, c("betavirus","betayear","sigmaepsilon"), n.iter=4000)
|*****| 100%
```

```
> gelman.diag(x2, autoburnin=FALSE)
```

Potential scale reduction factors:

	Point est.	Upper C.I.
betavirus[1]	1.01	1.02
betavirus[2]	1.00	1.01
betavirus[3]	1.01	1.02
betayear	1.01	1.01
sigmaepsilon	1.02	1.04

Multivariate psrf

1.01

```

> x2 <- coda.samples(m2, c("betavirus","betayear","sigmaepsilon","prob",
+                           "deathsrep","epsilon"), n.iter=2000)
|*****| 100%

> effectiveSize(x2[,c("betavirus[1]","betavirus[2]","betavirus[3]",
+                     "betayear","sigmaepsilon")])
betavirus[1] betavirus[2] betavirus[3]      betayear sigmaepsilon
    188.3363    224.2953    123.1827    136.7969    945.9479

```

Need more iterations for inference ...

```

> x2 <- coda.samples(m2, c("betavirus","betayear","sigmaepsilon","prob",
+                           "deathsrep","epsilon"), n.iter=10000)
|*****| 100%

> effectiveSize(x2[,c("betavirus[1]","betavirus[2]","betavirus[3]","
+                     "betayear","sigmaepsilon")])
betavirus[1] betavirus[2] betavirus[3]      betayear sigmaepsilon
      822.4306      980.2605      502.0411      589.0861      2472.3824

```

Effective sample sizes now adequate.

```
> summary(x2[,c("betavirus[1]","betavirus[2]","betavirus[3] ",
+               "betayear","sigmaepsilon")])
```

```
...
```

	Mean	SD	Naive SE	Time-series SE
betavirus[1]	-0.5122	0.5712	0.0028558	0.019973
betavirus[2]	1.1237	0.2479	0.0012397	0.007913
betavirus[3]	0.2233	0.3688	0.0018440	0.016826
betayear	-0.2605	0.4306	0.0021532	0.017559
sigmaepsilon	0.7040	0.1877	0.0009386	0.003780

```
...
```

	2.5%	25%	50%	75%	97.5%
betavirus[1]	-1.6614	-0.87322	-0.5112	-0.14831	0.6242
betavirus[2]	0.6216	0.96729	1.1228	1.28084	1.6215
betavirus[3]	-0.5238	-0.00676	0.2187	0.45343	0.9637
betayear	-1.0883	-0.53952	-0.2549	0.02211	0.5749
sigmaepsilon	0.4193	0.57087	0.6769	0.80514	1.1432

Note:

- ▶ `betayear` could plausibly be 0 (no time effect).
- ▶ `sigmaepsilon` is apparently not unduly restricted by its prior upper bound of 10.

Set up for chi-square discrepancy check, as before:

```
> probs <- as.matrix(x2)[, paste("prob[",1:nrow(ebola),"]", sep="")]
> deathsrep <- as.matrix(x2)[, paste("deathsrep[",1:nrow(ebola),"]", sep="")]
> Tchi <- numeric(nrow(deathsrep))
> Tchirep <- numeric(nrow(deathsrep))
> for(s in 1:nrow(deathsrep)){
+   Tchi[s] <- sum((ebola$Deaths - ebola$Cases*probs[s,])^2 /
+                 (ebola$Cases*probs[s,]*(1-probs[s,])))
+   Tchirep[s] <- sum((deathsrep[s,] - ebola$Cases*probs[s,])^2 /
+                     (ebola$Cases*probs[s,]*(1-probs[s,])))
+ }
```

```
> mean(Tchirep >= Tchi)
[1] 0.46925
```

No overdispersion problems this time.

Could also check  $\epsilon_i$ s for

- ▶ Non-normality
- ▶ Relationship to country most affected

These were checked (results not shown), and no problems found.



Investigate fatality rates of viruses for a “new” outbreak at average year (`mean(ebola$Year)`):

```
> ilogit <- function(x) 1/(1+exp(-x))

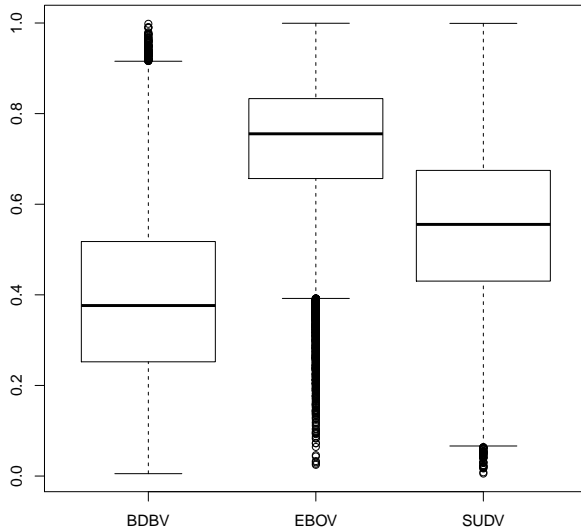
> betavirus <- as.matrix(x2)[, paste("betavirus[",1:3,"]", sep="")]

> sigmaepsilon <- as.matrix(x2)[, "sigmaepsilon"]

> newepsilon <- rnorm(length(sigmaepsilon), 0, sigmaepsilon)

> pBDBV.post <- ilogit(betavirus[,1] + newepsilon)
> pEBOV.post <- ilogit(betavirus[,2] + newepsilon)
> pSUDV.post <- ilogit(betavirus[,3] + newepsilon)

> boxplot(list(BDBV=pBDBV.post,EBOV=pEBOV.post,SUDV=pSUDV.post))
```



For example, the probability someone is more likely to die in a new EBOV outbreak than in a new SUDV outbreak:

```
> mean(pEBOV.post > pSUDV.post)
[1] 0.97415
```

The 2014 EBOV outbreak in DRC had 66 human cases and 49 deaths.

How many deaths does our model predict?

```
> betayear <- as.matrix(x2)[, "betayear"]

> newdeaths <- rbinom(nrow(betavirus), 66,
+                    ilogit(betavirus[,2] +
+                          betayear*(2014-mean(ebola$Year))/(2*sd(ebola$Year))
+                          + newepsilon))

> quantile(newdeaths, c(0.025, 0.5, 0.975))
 2.5%   50%  97.5%
   21    48    63
```