ADVANCED BAYESIAN MODELING

Ebola Data Example: Initial Model

Ebola Outbreaks Data

From 1976 to 2012, there were 17 major Ebola outbreaks.

Let

$$y_i \quad = \quad \text{number of deaths in outbreak } i\text{, out of } n_i \text{ human cases}$$

$$i = 1, \ldots, 17$$

Data includes year outbreak started, country most affected, and virus type.

(Note: Data excludes the West African Ebola virus epidemic, 2013–2016, which was of a disproportionate scale, so may not be comparable.)

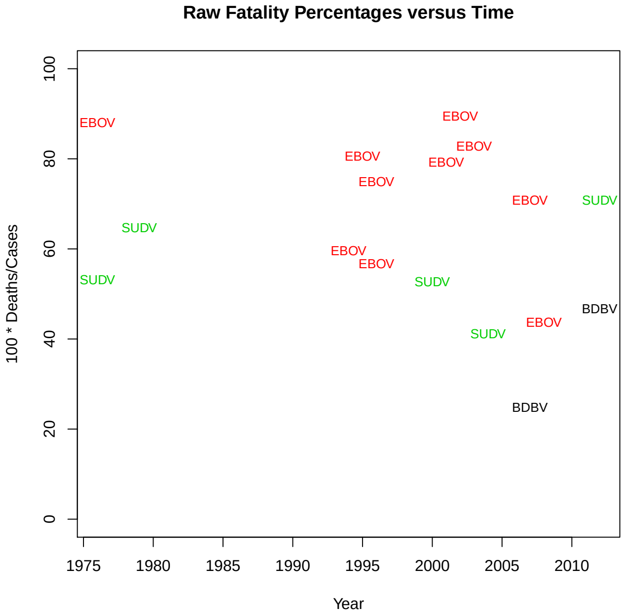
```
> ebola <- read.table("ebola.txt", header=TRUE)

> head(ebola)
  Year   Country Virus Cases Deaths
1 1976     Sudan  SUDV   284    151
2 1976 Zaire/DRC  EBOV   318    280
3 1979     Sudan  SUDV    34     22
4 1994     Gabon  EBOV    52     31
5 1995 Zaire/DRC  EBOV   315    254
6 1996     Gabon  EBOV    37     21

> levels(ebola$Virus)
[1] "BDBV" "EBOV" "SUDV"

> unclass(ebola$Virus)
 [1] 3 2 3 2 2 2 2 3 2 2 2 3 2 1 2 3 1
...
```

Note: Though the numbers of cases n_i are random, we will condition on these values.

Only the *fatality rates* (death probabilities) are of interest.

Fatality rates are thought to vary by virus type.

Fatality rates may also change with time (e.g., because of changes in medical care or evolution of viruses).


Raw Fatality Percentages versus Time
100 * Deaths/Cases
0
20
40
60
80
100
Year
1975
1980
1985
1990
1995
2000
2005
2010
EBOV
EBOV
EBOV
EBOV
EBOV
EBOV
SUDV
EBOV
SUDV
SUDV
EBOV
EBOV
SUDV
SUDV
BDBV
EBOV
SUDV
BDBV

Explanatory Variables

$$\begin{aligned} x_{i1},\, x_{i2},\, x_{i3} &= \text{indicators of ebolavirus type (BDBV, EBOV, SUDV)} \\ x_{i4} &= \text{year outbreak began (centered and scaled)} \end{aligned}$$

We allow the indicators of ebolavirus type to define the intercept (implicitly), so we do *not* center them.

As suggested previously, year is centered and scaled to have a sample standard deviation of 0.5.

Initial Model

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

$$\text{logit}(p_i) \quad = \quad X_i\beta \quad = \quad \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4}$$

We use the suggested scaled-t_1 priors for the coefficients in β (see BDA3, Sec. 16.3).

In ebola1.bug:

```
model {

  for (i in 1:length(deaths)) {
    deaths[i] ~ dbin(prob[i], cases[i])
    logit(prob[i]) <- betavirus[virus[i]] + betayear*yearscaled[i]

    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)

}
```

Notes:

- `logit(prob[i]) <- ...` is a valid way to specify the deterministic relationship involving the logit link function.
- `virus` will have to contain integer codes (1,2,3) for the virus types.
- `deathsrep` anticipates producing posterior predictive p-values for model checking.

Set up data and initial values for 4 chains:

```
> d1 <- list(deaths = ebola$Deaths,
+             cases = ebola$Cases,
+             virus = unclass(ebola$Virus),
+             yearscaled = as.vector(scale(ebola$Year, scale=2*sd(ebola$Year))))

> inits1 <- list(list(betavirus=c(10,10,10), betayear=10),
+                 list(betavirus=c(10,10,-10), betayear=-10),
+                 list(betavirus=c(10,-10,10), betayear=-10),
+                 list(betavirus=c(10,-10,-10), betayear=10))
```

Note: In logistic regression on scaled variables, coefficients with a magnitude of 10 are relatively extreme.

```
> library(rjags)
...

> m1 <- jags.model("ebola1.bug", d1, inits1, n.chains=4, n.adapt=1000)
...

> update(m1, 1000)  # burn-in
  |**************************************************| 100%

> x1 <- coda.samples(m1, c("betavirus","betayear"), n.iter=2000)
  |**************************************************| 100%
```

```
> gelman.diag(x1, autoburnin=FALSE)
Potential scale reduction factors:

             Point est. Upper C.I.
betavirus[1]          1          1
betavirus[2]          1          1
betavirus[3]          1          1
betayear              1          1

Multivariate psrf

1
```

```
> x1 <- coda.samples(m1, c("betavirus","betayear","prob","deathsrep"),
+                     n.iter=2000)
  |**************************************************| 100%

> effectiveSize(x1[,1:4])
betavirus[1] betavirus[2] betavirus[3]     betayear
    4052.981     4445.311     3696.059     3551.444
```

```
> summary(x1[,1:4])
...

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

                 Mean      SD  Naive SE Time-series SE
betavirus[1] -0.59766 0.15184 0.0016976       0.002384
betavirus[2]  1.27345 0.06778 0.0007579       0.001019
betavirus[3]  0.04799 0.07706 0.0008616       0.001268
betayear     -0.31884 0.09981 0.0011159       0.001695

2. Quantiles for each variable:

                2.5%       25%      50%      75%   97.5%
betavirus[1] -0.8987 -0.699800 -0.59768 -0.49633 -0.3030
betavirus[2]  1.1433  1.227456  1.27341  1.31792  1.4102
betavirus[3] -0.1020 -0.005069  0.04908  0.09929  0.1997
betayear     -0.5185 -0.383386 -0.31918 -0.25137 -0.1266
```

For checking overdispersion, first extract the samples of the fitted probabilities p_i and the replicate responses y_i^{rep}:

```
> probs <- as.matrix(x1)[, paste("prob[",1:nrow(ebola),"]", sep="")]

> deathsrep <- as.matrix(x1)[, paste("deathsrep[",1:nrow(ebola),"]", sep="")]
```

Now compute samples of chi-square discrepancy and replicated chi-square discrepancy:

```
> 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
```

Quite substantial evidence of a problem, probably overdispersion.

We need a better model that takes account of this ...