STAT 8700 Homework 11

Brian Detweiler

Monday, December 12th

```
##
## *******************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *******************************************************************
```

1. The file `planes.txt` contains 30 observations of 4 variables. You can read this into R using `read.table("planes.txt", header=T)`. The data is from 30 Air Force missions during the Vietnam war. The 4 variables are as follows:

y **is the the number of damaged locations of the aircraft;**

$x1$ **is the type of aircraft, 0 for A4, 1, for A6;**

$x2$ **is the aircraft bomb load in tons;**

$x3$ **is the total months of aircrew experience.**

Model y **in JAGS using Poisson regression with a log link function. Use DIC to determine which of the three explanatory variables should be included in your model. Once you have identified the the best model, use it calculate 95% prediction intervals for the amount of damage for both A4 and A6 planes, with a crew with minimal, average, and maximal experience, for a minimal, average, and maximal bomb load.**

```r
planes <- read.table('planes.txt', header=TRUE)
head(planes)
```

```
##   Observation y x1 x2   x3
## 1           1 0  0  4 91.5
## 2           2 1  0  4 84.0
## 3           3 0  0  4 76.5
## 4           4 0  0  5 69.0
## 5           5 0  0  5 61.5
## 6           6 0  0  5 80.0
```

Let's try the zero-inflated model first:

```r
fileName <- "Assignment_11.1"

modelString ="
model {

  for (j in 1:count) {
    y[j] ~ dpois(theta[j] * (1 - U))
    theta[j] <- exp(beta1 * x1[j] + beta2 * x2[j] + beta3 * x3[j])
  }

  U ~ dbern(pi0)
  pi0 ~ dbeta(1, 1)

  beta1 ~ dnorm(0, 1.0E-4)
  beta2 ~ dnorm(0, 1.0E-4)
  beta3 ~ dnorm(0, 1.0E-4)

}
"

writeLines(modelString, con=fileName)

planes.model.zero.infl = jags.model(file=fileName,
                                    data=list(y=planes$y,
                                              count=length(planes$y),
                                              x1=planes$x1,
                                              x2=planes$x2,
                                              x3=planes$x3),
                                    n.chains=4)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 30
##    Unobserved stochastic nodes: 5
##    Total graph size: 294
##
## Initializing model
```

```r
update(planes.model.zero.infl, n.iter=50000)

planes.DIC.zero.infl <- dic.samples(model = planes.model.zero.infl, n.iter = 200000, thin = 50)
planes.DIC.zero.infl
```

```
## Mean deviance:  82.95
## penalty 3.044
## Penalized deviance: 85.99
```

```r
planes.samples.zero.infl <- coda.samples(model = planes.model.zero.infl,
                                         variable.names = c("y", "beta1", "beta2", "beta3"),
                                         n.iter = 200000,
                                         thin = 50)

diagMCMC(planes.samples.zero.infl)
```

Now let's try a non-zero-inflated Poisson regression:

```r
fileName <- "Assignment_11.2"

modelString ="
model {

  for (j in 1:count) {
    y[j] ~ dpois(theta[j])
    theta[j] = exp( eta[j] )
    eta[j] = beta1 * x1[j] + beta2 * x2[j] + beta3 * x3[j]
  }

  beta1 ~ dnorm(0, 1.0E-4)
  beta2 ~ dnorm(0, 1.0E-4)
  beta3 ~ dnorm(0, 1.0E-4)

  a4.min.min ~ dpois(exp(beta2 * min(x2)  + beta3 * min(x3)))
  a4.avg.min ~ dpois(exp(beta2 * mean(x2) + beta3 * min(x3)))
  a4.max.min ~ dpois(exp(beta2 * max(x2)  + beta3 * min(x3)))
  a4.min.avg ~ dpois(exp(beta2 * min(x2)  + beta3 * mean(x3)))
  a4.avg.avg ~ dpois(exp(beta2 * mean(x2) + beta3 * mean(x3)))
  a4.max.avg ~ dpois(exp(beta2 * max(x2)  + beta3 * mean(x3)))
  a4.min.max ~ dpois(exp(beta2 * min(x2)  + beta3 * max(x3)))
  a4.avg.max ~ dpois(exp(beta2 * mean(x2) + beta3 * max(x3)))
  a4.max.max ~ dpois(exp(beta2 * max(x2)  + beta3 * max(x3)))

  a6.min.min ~ dpois(exp(beta1 + beta2 * min(x2)  + beta3 * min(x3)))
  a6.avg.min ~ dpois(exp(beta1 + beta2 * mean(x2) + beta3 * min(x3)))
  a6.max.min ~ dpois(exp(beta1 + beta2 * max(x2)  + beta3 * min(x3)))
  a6.min.avg ~ dpois(exp(beta1 + beta2 * min(x2)  + beta3 * mean(x3)))
  a6.avg.avg ~ dpois(exp(beta1 + beta2 * mean(x2) + beta3 * mean(x3)))
  a6.max.avg ~ dpois(exp(beta1 + beta2 * max(x2)  + beta3 * mean(x3)))
  a6.min.max ~ dpois(exp(beta1 + beta2 * min(x2)  + beta3 * max(x3)))
  a6.avg.max ~ dpois(exp(beta1 + beta2 * mean(x2) + beta3 * max(x3)))
  a6.max.max ~ dpois(exp(beta1 + beta2 * max(x2)  + beta3 * max(x3)))
}
"

writeLines(modelString, con=fileName)

planes.model.pois = jags.model(file=fileName,
                               data=list(y=planes$y,
                                         count=length(planes$y),
                                         x1=planes$x1,
                                         x2=planes$x2,
```

```r
                                          x3=planes$x3),
                               n.chains=4)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 30
##    Unobserved stochastic nodes: 21
##    Total graph size: 293
##
## Initializing model
```

```r
update(planes.model.pois, n.iter=50000)


planes.DIC.pois <- dic.samples(model = planes.model.pois, n.iter = 200000, thin = 50)

planes.DIC.pois
```

```
## Mean deviance:  82.86
## penalty 2.972
## Penalized deviance: 85.83
```

```r
diffdic(planes.DIC.pois, planes.DIC.zero.infl)
```

```
## Difference: -0.1592699
## Sample standard error: 0.0354502
```

This seems to have a slightly better fit than the zero-inflated Poisson model.

```r
planes.samples.pois <- coda.samples(model = planes.model.pois,
                                    variable.names = c("beta1",
                                                       "beta2",
                                                       "beta3",
                                                       "a4.min.min",
                                                       "a4.avg.min",
                                                       "a4.max.min",
                                                       "a4.min.avg",
                                                       "a4.avg.avg",
                                                       "a4.max.avg",
                                                       "a4.min.max",
                                                       "a4.avg.max",
                                                       "a6.min.min",
                                                       "a6.avg.min",
                                                       "a6.max.min",
                                                       "a6.min.avg",
                                                       "a6.avg.avg",
                                                       "a6.max.avg",
                                                       "a6.min.max",
                                                       "a6.avg.max",
```

```
                                                   "a6.max.max"),
                                      n.iter = 200000,
                                      thin = 50)

planes.samples.pois.M <- as.matrix(planes.samples.pois[,"beta1"])
```

Our 95% prediction intervals for each situation, A4 or A6 aircraft, with minimal, average, or maximal bomb load, and minimal, average, or maximal crew experience.

```
summary(planes.samples.pois)
```

```
##
## Iterations = 251050:451000
## Thinning interval = 50
## Number of chains = 4
## Sample size per chain = 4000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                 Mean       SD  Naive SE Time-series SE
## a4.avg.avg  0.85681 0.975061 7.709e-03      7.807e-03
## a4.avg.max  0.46137 0.708812 5.604e-03      5.604e-03
## a4.avg.min  1.45100 1.286158 1.017e-02      1.026e-02
## a4.max.avg  2.23937 1.994260 1.577e-02      1.596e-02
## a4.max.min  3.91188 3.298339 2.608e-02      2.676e-02
## a4.min.avg  0.46763 0.691453 5.466e-03      5.466e-03
## a4.min.max  0.25500 0.513192 4.057e-03      4.097e-03
## a4.min.min  0.77975 0.903350 7.142e-03      7.142e-03
## a6.avg.avg  1.57862 1.316454 1.041e-02      1.032e-02
## a6.avg.max  0.83981 0.975237 7.710e-03      7.635e-03
## a6.avg.min  2.68763 1.744703 1.379e-02      1.391e-02
## a6.max.avg  3.71731 2.080486 1.645e-02      1.625e-02
## a6.max.max  1.89675 1.450550 1.147e-02      1.146e-02
## a6.max.min  6.42675 3.162931 2.501e-02      2.464e-02
## a6.min.avg  0.92563 1.039067 8.215e-03      8.490e-03
## a6.min.max  0.50200 0.766914 6.063e-03      6.189e-03
## a6.min.min  1.53494 1.352320 1.069e-02      1.083e-02
## beta1       0.64429 0.488012 3.858e-03      3.969e-03
## beta2       0.14583 0.053099 4.198e-04      4.428e-04
## beta3      -0.01726 0.005057 3.998e-05      4.067e-05
##
## 2. Quantiles for each variable:
##
##                2.5%      25%      50%      75%     97.5%
## a4.avg.avg  0.00000  0.00000  1.00000  1.00000  3.000000
## a4.avg.max  0.00000  0.00000  0.00000  1.00000  2.000000
## a4.avg.min  0.00000  0.00000  1.00000  2.00000  4.000000
## a4.max.avg  0.00000  1.00000  2.00000  3.00000  7.000000
## a4.max.min  0.00000  2.00000  3.00000  5.00000 12.000000
## a4.min.avg  0.00000  0.00000  0.00000  1.00000  2.000000
## a4.min.max  0.00000  0.00000  0.00000  0.00000  2.000000
## a4.min.min  0.00000  0.00000  1.00000  1.00000  3.000000
```

```
## a6.avg.avg  0.00000  1.00000  1.00000  2.00000  5.000000
## a6.avg.max  0.00000  0.00000  1.00000  1.00000  3.000000
## a6.avg.min  0.00000  1.00000  2.00000  4.00000  7.000000
## a6.max.avg  0.00000  2.00000  3.00000  5.00000  8.000000
## a6.max.max  0.00000  1.00000  2.00000  3.00000  5.000000
## a6.max.min  1.00000  4.00000  6.00000  8.00000 13.025000
## a6.min.avg  0.00000  0.00000  1.00000  1.00000  3.000000
## a6.min.max  0.00000  0.00000  0.00000  1.00000  2.000000
## a6.min.min  0.00000  1.00000  1.00000  2.00000  5.000000
## beta1      -0.29233  0.31208  0.63532  0.97063  1.621918
## beta2       0.04268  0.10992  0.14503  0.18111  0.251819
## beta3      -0.02750 -0.02063 -0.01712 -0.01374 -0.007761
```

```
planes.df <- as.data.frame(as.matrix(planes.samples.pois))
planes.df <- planes.df %>% select(-beta1, -beta2, -beta3)

planes.df.melt <- melt(planes.df)

ggplot(planes.df.melt, aes(x=variable, y=value)) +
  geom_boxplot() +
  coord_flip() +
  labs(x = "Aircraft.BombLoad.CrewExperience", y = "Damaged Locations")
```