Course EPIB-668 - Bayesian Analysis in Medicine

Assignment 5

1. The FIM is a measure of functional independence, often used in the emergency room following an accident or trauma. Higher values of the FIM indicate better functioning, with 126 being considered as "normal". In this question we will consider the effect of alcohol on the FIM. This is an interesting question, since it is not clear whether alcohol will have a positive (e.g., more relaxed during an accident may lead to less injuries) or negative (e.g., worse accidents if under the influence) effect on the FIM, on average.

(a) The course web site has a data set called fimdata.txt. Download this file somewhere to your hard disk. Open R and read in the data set (using a command such as

```
fim.dat <-read.table(file, header=T),
```

where file points to where you stored the data set, such as "c://temp//fimdata.txt"). The variables include age in years, sex (1=male, 0=female), fim, and alcohol level on arrival at the emergency room. Print out some descriptive statistics for these variables, for example using the R command summary(fim.dat). Print out plots of age versus fim and alcohol versus fim. Do you see any trends?

(b) To use the data set in WinBUGS, you will need to add []'s after each variable name. Open WinBUGS, and cut and paste in the data set you saved. Change the first line to

```
sex[]  age[] fim[]  alcohol[]
```

Run a simple linear regression in WinBUGS, with fim as the dependent variable, and age, sex, and alcohol level. Report your results, and provide an interpretation for each beta coefficient.

(c) It is easy to make predictions for any variable combination using WinBUGS, simply by adding a single line to the program for each prediction you want to make. For example, if you want to make a fim prediction for a male aged 50 with an alcohol level of 80, add a line like:

```
pred.fim.male.50.80 <- alpha + beta.sex*1 + beta.age*50 + beta.alc*80
```

and then monitor the new parameter you created, pred.fim.male.50.80. Create predictions for males aged 30 and alcohol level of 85 and for females aged 50 with alcohol level of zero. Create another variable that monitors the difference between these two groups, with a command such as

```
diff <- pred.fim.male.30.85 - pred.fim.female.50.0
```

Run a program with these additional lines, report the outcomes, and comment of the difference between the two sets of predictions you made.

2. One concern one should always have after running a model is whether or not it fits the data well.

(a) We can calculate Bayesian residuals by making a prediction for each individual in the data set, and taking the differences between observed and predicted results. For example, if your model is

```
fim[i] <- alpha + beta.sex*sex[i] + beta.age*age[i]
       + beta.alc*alcohol[i]
```

then residuals can be formed by adding the lines

```
fim.pred[i] <- alpha + beta.sex*sex[i] + beta.age*age[i]
       + beta.alc*alcohol[i]
residual[i] <- fim[i] - fim.pred[i]
```

Run this model, monitoring the residual variable. In the WinBUGS output, the column for the mean of the residuals roughly corresponds to the usual frequentist residuals. Looking down this column, can you tell if the model fits well?

(b) Because each residual is looked at separately, it is difficult to get an overall picture of fit. This problem can be remedied by taking a random selection of residuals, and mixing them all together. This can be done by adding lines such as:

```
for(i in 1:60)
{
p[i] <- 1/60
}
choose ~ dcat(p[])
residual.mixture <- residual[choose]
```

Now monitoring this residual.mixture gives you a good idea of the residuals from all subjects mixed together. You can look at the residual plot using the density option, or just look at the usual WinBUGS statistics on residual.mixture. Does the model seem to fit well?

Note: Bayesian residuals account for imperfections in the model, unlike frequentist residuals, which simply compare observed values with values predicted from the mean model.

3. CRP is a measure of inflammation, of interest to cardiologists for predicting future heart attacks. From the course web site, download the CRP data set (crpdata.txt), which is already in a format useable in WinBUGS. The data represent six CRP values from each of 15 subjects, followed over three years, with measurements taken every six months.

(a) Plot these data in R. You can use read.table to get the data into R, but you will need to erase the []'s that follow each variable name (but leave the number inside the []'s), and take the word "END" away. To plot the data, we would like a single line to represent the data from each subject, so want a graph with 15 lines across it, one for each subject. R code such as the following could be used.

```
> plot(1:6,crp.dat[1,], type="l", ylim=c(0,15), xlab="time", ylab="crp")
> for (i in 2:15){ points(1:6,crp.dat[i,], type="l", lty=i)}
```

Comment on the results. Is there much variability of CRP values over time within each subject? Is there much between-subject variability?

(b) Assume the following two-level hierarchical structure for the data: Within each subject, assume that the six CRP values are independent over time, and follow a normal distribution with mean specific to that individual, but with a variance (precision in WinBUGS) parameter common to all subjects. Further, assume that the subject-specific means over time follow a common normal distribution across subjects. Using non-informative priors for all unknown parameters (15 means for each subject, one overall mean across all subjects, and two variance/precision parameters, one for the variability within each subject over the time, and the other for the between subject means), write and run a WinBUGS program, monitoring all unknown parameters. Print out your program and basic statistics. Comment on whether the results coincide with your expectations from part (a).

4. The data set infant.dat.txt contains two variables: X is the gestational age of infants (in weeks) at the time of birth and Y is a yes/no (1/0) variable indicating whether the infant was breast feeding at the time of release from hospital or not. The data are already in WinBUGS format. Analyse these data in WinBUGS using a standard logistic regression model. Use non-informative priors (exact form is your choice) and provide statistics for all model parameters. Your program should also provide the marginal posterior density for the odds ratio of the effect of gestational age on the probability of being breastfed, as well as the predicted probability of being breastfed for an infant with gestational age of 32 weeks.

5. While so far we have seen standard and hierarchical linear and logistic regression models, there is absolutely nothing special about these models as far as WinBUGS is concerned. In fact, it is just as easy to fit a nonlinear model as it is to fit a linear model, as we are about to see, using the data set in the file nonlinear.dat.txt on the course web site.

(a) Read the file into R, and plot x versus y and z versus y. Note the highly non-linear character of the relationship between the independent variables (x and z) and the dependent variable y.

(b) Fit the following non-linear equation to the data (suppose the specific form came from the theory associated with the experiment that gave rise to the data):

$$y = sin(a * (x^2)) * z^b + sqrt(z^c)$$

The parameters of the model are a, b, and c, which are assumed unknown. Assume non-informative prior distributions for these parameters, and assume an error model for the model which is normal, with zero mean and unknown variance, which is also given a non-informative prior. Thus, model of the WinBUGS program should look something like:

```
model
{
for (i in 1:100)
{
y.mean[i] <- sin(a*(x[i]*x[i]))*pow(z[i],b) + sqrt(pow(z[i],c))
y[i] ~ dnorm(y.mean[i], tau)
}
a ~ dnorm(0, 0.01)
b ~ dnorm(0, 0.01)
c ~ dnorm(0, 0.01)
tau <- 1/(sigma*sigma)
sigma ~ dunif(0.001, 10)
# inits
list(a=.2, b=3, c=4, sigma=1)
}
```

Run a WinBUGS program that includes code like the above as its main model, and estimate a, b, and c. Use the starting values as given. Because the surface is highly multi-model, WinBUGS needs to be started near the true maximum. [There are methods beyond the scope of this course for dealing with such problems in real practice, here I just wanted to demonstrate the range of models that can be fit with WinBUGS, as a crude example.] Note in particular that this model was as easy to run as any standard linear model! How would you estimate this model using non-Bayesian methods?