Bayesian Statistics - Exercise 2

Objective

In this second exercise you will learn more about Bayesian posterior predictive checks for model assumptions, by conducting one in R with output from JAGS using the R package RJAGS.

Help files

If you need help, the JAGS user manual or the rjags reference manual can be consulted. If you are stuck with a specific error message, a Google search may turn up possible causes and solutions.

The data (same as in Exercise 1)

Several studies suggest that cognitive behavioral therapy is an effective treatment for Post-Traumatic Stress Disorder (PTSD) in male veterans. Suppose that you did a study to compare Prolonged Exposure (PE), a type of cognitive behavioral therapy, with Present-Centered therapy (PC), a supportive intervention. It was a randomized controlled trial, where 284 veterans suffering from PTSD were assigned randomly to receive either PE or PC. The outcome measure of interest was loss of diagnosis (LD), a dichotomous variable. The resulting data are displayed in Table 1.

Table 1: PTSD data

	Type of Intervention	
	PE	PC
Loss of Diagnosis	58	40
Total Men Treated	141	143

In Exercise 1, you analyzed this data to examine whether PE is an effective treatment. Now, you want to check the feasibility of model assumptions before publishing your results.

A. Recognizing the model assumption(s).

Every model comes with assumptions about the research design and the data. In this case, you used two binomial models for the number of persons that Lost their Diagnosis, estimating the probability of recovery separately for the PE and PC group (and the Relative Risk measure to compare them). As you may recall, the binomial distribution concerns the number of successes in a sequence of n independent Bernoulli trials with success probability p. What does this assume about your research design and/or data? Can you think of plausible reasons why this assumption might not hold?

B. Writing the equivalent model for the raw (individual) data.

Since you used two separate binomial models, any model assumption should be tested separately for each group, as well. But to do this, you need the raw data instead of only the summary statistics (i.e., the total numbers of successes). The file `Exercise 2 - Data.txt` contains the dichotomous score (1=Loss of Diagnosis, 0=no recovery) for each person in the study. By comparing this raw data set to many model-predicted (replicated) 'data sets' you can inspect whether the empirical data violate a model assumption. For this purpose, start by writing a model file that repeats

the same analysis of Exercise 1 (with uninformative priors), but now using the raw data as input instead of the total numbers of recoveries. You can use `Exercise 2 - Model_template.txt` as a template.
Tip: Look again at your model file for Exercise 1, and consult the list of available distributions in JAGS in Chapters 6, 7 and 9 of the JAGS user manual version 4.3.0.

C. Running the analysis, checking the results against Exercise 1.

When you are done specifying the model, run the analysis and compare the parameter estimates with those in Exercise 1. If your model is correct and you used the same prior specifications, the results should be similar.

D. Specifying test statistics for the empirical data.

One way of testing whether a binomial distribution fits the data is to check whether the proportion of recoveries (LD) in the first half of each group is equal to the proportion in the second half of that group. If it isn't, do you see what this could imply about your study?

Calculate this test statistic measuring the discrepancy between the model and the empirical data in R with the following statements :

```
# test statistic for the PE condition
proportion.half1.PE <- sum(dat$LD.PE[1:70])/70
proportion.half2.PE <- sum(dat$LD.PE[71:141])/71
diff.PE <- proportion.half1.PE - proportion.half2.PE
# test statistic for the PC condition
proportion.half1.PC <- sum(dat$LD.PC[1:71])/71
proportion.half2.PC <- sum(dat$LD.PC[72:143])/72
diff.PC <- proportion.half1.PC - proportion.half2.PC
```

Note that these two test statistics are solely based on the empirical data, not on model estimates. Therefore, `diff.PE` and `diff.PC` are fixed quantities, that are only calculated once.

E. Calculating the test statistics for model-predicted (replicated) data sets.

To conduct a posterior predictive check, you need 'replicated' data sets that are generated in each iteration of the sampler using the current model estimates. These data sets represent a sampling distribution under the null hypothesis that the model is true (but taking into account uncertainty about the exact parameter values).

To sample datasets from the posterior predictive distribution, you first have to extract the stored parameter estimates for each iteration from the samples object with the following statements:

```
# Extract the parameter estimates
theta.PE.chain1 <- samples[[1]][,"theta.PE"]
theta.PE.chain2 <- samples[[2]][,"theta.PE"]
theta.PC.chain1 <- samples[[1]][,"theta.PC"]
theta.PC.chain2 <- samples[[2]][,"theta.PC"]
```

Then you can use the parameter estimate values to sample replicated datasets:

```
# Load LaplacesDemon library for rbern() function
library(LaplacesDemon)

# Storage room (each row is a replicated dataset) :
replicated.PE.chain1 <- array(data = NA, dim = c(length(theta.PE.chain1), dat$n.PE))
replicated.PE.chain2 <- array(data = NA, dim = c(length(theta.PE.chain2), dat$n.PE))
replicated.PC.chain1 <- array(data = NA, dim = c(length(theta.PC.chain1), dat$n.PC))
replicated.PC.chain2 <- array(data = NA, dim = c(length(theta.PC.chain2), dat$n.PC))

 # Sample replicated datasets
 # For each parameter estimate...
 for(t in 1:length(theta.PE.chain1)) {
 # ... sample a replicated dataset by sampling n times from the Bernoulli distribution
 # with as probability of success on each trial the parameter estimate
 replicated.PE.chain1[t,] <- rbern(n = dat$n.PE, prob = theta.PE.chain1[t])
 replicated.PE.chain2[t,] <- rbern(n = dat$n.PE, prob = theta.PE.chain2[t])
 replicated.PC.chain1[t,] <- rbern(n = dat$n.PC, prob = theta.PC.chain1[t])
 replicated.PC.chain2[t,] <- rbern(n = dat$n.PC, prob = theta.PC.chain2[t])
}
```

Before continuing with F, you still have to write statements that calculate the test statistics for the replicated data sets, so that you can compare the (fixed) test statistics for the empirical data sets with the distribution of posterior predicted measures. Note: so far, we have repeated every piece of code for both chains. However, this is not very efficient and will result in a lengthy code. If you feel like it, you might want to change the code for the posterior predictive check in R to calculate the results for both chains within one `for` loop.

F. Adding code to obtain posterior predictive p-values for the two groups.

Consider whether you want one-sided or two-sided tests in this situation. What do you think makes more sense here?

Include statements in R to obtain the posterior predictive p-value for each group. Hint: the actual p-values are only obtained by looking at the means of the 0/1 variables (every p-value is a proportion!). In each iteration of the sampler, the test statistic for a replicated data set either is, or is not, more extreme than for the empirical data set. For a logical test you can use the `ifelse` function which is defined as follows: `ifelse(test, if yes, if no)`.

G. Interpreting the posterior predictive p-values.

Look at the means of the 0/1 variables you defined in the previous step. What do the results for the posterior predictive p-value (ppp) tell you about your analysis? And how does this posterior predictive check relate to classical (frequentist) hypothesis testing? Describe the similarities and differences.