Course EPIB-621 - Data Analysis for the Health Sciences

Assignment 5

1. Recall again the data set used for assignment 4 called drugfree.txt. To remind you the variables contained in this data set are described in the table below:

Description	Code	Variable Name
Age at Enrollment	Years	age
Beck Depression Score at Admission	0.000-54.000	beck
IV Drug Use History at Admission	1 = Never, 2 = Previous 3 = Recent	ivhx
Number of Prior Drug Treatments	0-40	ndrugtx
Subject's Race	0 = White 1 = Other	race
Treatment Randomization Assignment	0 = Short 1 = Long	treat
Remained Drug Free for 12 Months	1 = Remained Drug Free 0 = Otherwise	drug.free

We previously used this data set to crudely investigate confounding, but now that we have learned about the bic.glm function, we can investigate this use further, as well as see what final model(s) can be used for best predictions.

(a) Run bic.glm on this data set, and report what variables were selected to be in the best model. Recall that you need to load the BMA program first, before running the bic.glm command. Remember also to declare the inhx variable as a factor before you run the regressions. Report the analysis summary, which includes the best five models, and the model probabilities associated with each of these five models.

(b) Output the maximum likelihood estimators from the top 14 models along with

the SEs from these models. Looking down the columns for each model, do you see any evidence for confounding? If so, report on which variables may be confounded.

(c) Report your "best" (i.e., model averaged) prediction for someone who is aged 30, has a beck depression score of 10, has a recent history of IV drug use, is white, no prior drug treatment, and had a short treatment scheme.

2. The risk of a heart attack increases with age, but varies from country to country depending on local habits. The data set heart.txt provides the age and heart attack data (heart = yes/no = 1/0). The data are structured such that the first 100 subjects are from country 1, the second 100 are from country 2, and so on, so that the last 100 are from country 5. Therefore, no country variable needs to be defined. Use WinBUGS to create a hierarchical logistic regression model. The first level (individual patient level) regression should depend on a random intercept coefficient which is different for each country, and a fixed age coefficient, which is considered as constant across countries. All prior distributions should be normal, including the hierarchial distribution on the intercepts.

(a) Run this model in WinBUGS, and report the results from all coefficients, including the random effect (hierarchical) intercepts.

(b) Create lines such as

```
diff12 <- step(alpha[1]- alpha[2])
diff13 <- step(alpha[1]- alpha[3])
```

and so on to estimate the probability that one country's rate (after adjusting for the effect of age) is different from another country's rates. Since we have five countries, you will need to add 10 such lines. Run the program, and report these probabilities. [To save running time, you can run parts (a) and (b) as one program.]

3. There exists drugs or other treatments that are known to work for some subjects but not for others. Sometimes the reason for this can be ascertained, but not always. This can happen, for example, if an as yet undiscovered gene affects response to the substance. In clinical trials of these substances, any effects are typically reported as an average over responders and non-responders.

One example of a substance that seems to work for some subjects but not others is calcium supplementation for reduction in blood pressure; some subjects seem to respond, some do not. While the reason is not at this time known, we will suppose that there is an undiscovered gene responsible for this effect.

Suppose that a multi-centre clinical trial will be carried out to estimate the effect of calcium supplementation on lowering blood pressure. Two towns will participate, A and B. Suppose that the gene tends to be present in town A, but not town B.

Download the data file "calcium.txt" from the course web page. [While there, you might want to also download calcium.missing.txt, which will be used later.]

(a) Using WinBUGS, do a simple linear regression of blood pressure reduction on calcium intake. Report the average effect of calcium supplements, with 95% credible interval. Note that once you note the pattern of which subjects come from which town, you do not need the town variable.

(b) Do separate linear regressions within each town. Compare the effects of calcium in town A versus town B. Note that you can do all of this within a single WinBUGS program, by simply looping twice, once over the first 300 subjects, all from town A, and then over the next 300 subjects, all from town B. By creating a new parameter such as

```
beta.calcium.a - beta.calcium.b
```

you can directly monitor the difference in effects of calcium supplementation between the two towns.

We will now repeat the analysis of this clinical trial, but using data sets calcium.missing.txt, which contains some missing data.

(c) Using only the first 400 subjects in the database (i.e., those without any missing data), use a linear regression model to estimate the effect of calcium supplementation. Compare your answer to that obtained in part (a).

(d) Now use multiple imputation to adjust your answer in part (c). Using all subjects in the data set calcium.missing.txt, impute the missing data on the effects. Use a separate prediction equation for each of town A and town B. Now compare your answer to both parts (a) and (c). Has multiple imputation removed the bias in the estimated coefficient (which represents the average effect of calcium supplementation in these two towns)?

4. Generate a simulated linear regression data set in R that follows the following model (sample size = 100):

$$y = 2 + 5 * x, \quad \sigma = 1, \quad x \sim normal(0, 1)$$

To do this, use lines such as:

```
x <- round(rnorm(100, mean=0, sd=1),2)
y <- round(rnorm(100, mean = 2 + 5*x, sd=1),2)
```

Note that since we are using random numbers, everyone in the class will be using a slightly different data set. I rounded everything to 2 decimal places, which makes for cleaner data sets without losing too much precision.

(a) Plot x versus y.

(b) Use R to run a standard linear regression of x versus y. Provide the estimates and 95% confidence intervals for the intercept and slope (see R class notes if you forget how to do this, and recall that approximate 95% intervals can be derived from the point estimates ± 1.96 times the standard error for each parameter). Are they close to their theoretical values (of 2 and 5, respectively)?

(c) Now we will add some measurement error to the x values. In particular, we will create a measurement error version of x using the R command

```
x.error <- round(rnorm(100, mean=x, sd=2),2 )
```

Note that the measurement error version of x is centered at the true value of x, but has random noise about the observation. This is typical of measurement error seen when data are generated by an unbiased but imprecise measuring tool. Plot $x.error$ versus y, and note any differences from your plot in part (a).

(d) Rerun the linear regression again, but this time using $x.error$ rather than x. Compare the results (point estimates and confidence intervals) you obtain here with those obtained in part (b), and note any differences.

(e) Before leaving R, save your data sets for use in WinBUGS in problem 5. To do this, use commands such as:

```
x.list <- list(x=x, y=y)
xerror.list <- list(x.error = x.error, y=y)
dput(x.list, file = "c://temp//x.txt")
dput(xerror.list, file = "c://temp//xerror.txt")
```

You will use the first data set in the (a) of question 5, and the second data set in parts (b) and (c) of question 5.

5. In this question we will analyse the same two data sets as were used in question 4, but now using WinBUGS, with and without correcting for possible measurement error.

(a) Run a straightforward WinBUGS program for the linear regression of x versus y (see class notes of simple WinBUGS programs if you do not recall how to do this). Provide the point estimates and 95% credible intervals for the intercept and slope. Compare these to your estimates in part (b) of question 4 (they should be quite similar).

(b) Repeat part (a), but now using $x.error$ versus y. Provide the point estimates and 95% credible intervals for the intercept and slope. Compare these to your estimates in part (d) of question 4 (again, they should be quite similar).

(c) Now, we will modify the simple linear regression model to account for any measurement error. To the basic linear regression model (from part (a), NOT part (b), because we want to estimate the true relationship with x, not the one with measurement error variable $x.error$!), add a line such as:

```
x.error[i] ~ dnorm(x[i], tau.error)
```

You will also need to add a line for the prior for tau.error, and for x[i]. As usual, we will define tau.error in terms of sigma.error, and put a uniform prior on sigma. Use the following lines:

```
tau.error <- 1/(sigma.error*sigma.error)
sigma.error ~ dunif(1, 5)
```

to indicate that it is known that the measurement error variance is between 1 and 5 (real value, recall, was SD=2).

Run this model, and report the point estimates and 95% credible intervals for the intercept and slope. Compare these to your estimates in part (b) of question 5 ... has the model correctly adjusted for the measurement error?