

**Bayesian Statistics, 22S:138**  
Lab 5, Oct. 21, 2004  
Linear Regression Models in WinBUGS

**Simple linear regression model**

In the 1840s and 1850s a Scottish physicist, James D. Forbes, wanted to be able to estimate altitude above sea level from measurement of the boiling point of water. He knew that altitude could be determined from atmospheric pressure, measured with a barometer, with lower pressures corresponding to higher altitudes. In the experiments discussed here, he studied the relationship between pressure and boiling point. His interest in the problem was motivated by the difficulty in transporting the fragile barometers of the 1840s. Measuring the boiling point would give travelers a quick way of estimating altitudes.

Forbes reported data collected by Dr. Joseph Hooker in the Himalaya Mountains. After choosing each location, Hooker assembled his apparatus and measured pressure and boiling point. Pressure measurements were recorded in inches of mercury, adjusted for the difference between the ambient air temperature when he took the measurements and a standard temperature. Boiling point was measured in degrees Fahrenheit. Forbes' theory suggested that over the range of observed values the graph of boiling point versus the *logarithm* of pressure yields a straight line. Following Forbes, we take logs to the base 10, although the base of the logarithms is irrelevant for the statistical analysis. Since the logs of the pressures do not vary much, we multiply all the values of  $\log(\text{pressure})$  by 100 to avoid studying very small numbers without changing the major features of the analysis.

A dataset consisting of  $n = 31$  pairs of measurements on `temp` and `pres` as well as the transformed pressures, is posted as "hooker2.dat" under "Datasets" on the course web page.

The following code is also posted under "Datasets" under the name *line.txt*.

Model 1

```
model
{
  temp.bar <- mean(temp[])
  for (i in 1:N) {
    pres[i] ~ dnorm( mu[i], tau)
    mu[i] <- alpha + beta * (temp[i] - temp.bar)
    logpres[i] ~ dnorm( 0, .001)    # have to account for unused vbl
  }
  sigma <- 1/sqrt(tau)
  alpha ~ dnorm(0, 1.0E-6)
  beta ~ dnorm(0, 1.0E-6)
  tau ~ dgamma(0.001, 0.001)
}
```

```
list(N=31)

inits
list(alpha = 0, beta = 0, tau = 1)
```

Copy the code into one WinBUGS window and the dataset into another. Add column headings to the dataset, and add the word END in all caps, followed by a carriage return at the bottom, so it looks like this in the window:

```
temp[] pres[] logpres[]
210.8 29.211 146.555
210.2 28.559 145.574
208.4 27.972 144.672
202.5 24.697 139.264
. . .
. . .
END
```

We first will we regress `pres` on `temp`.

We are interested in estimating:

- the regression coefficients
- the standard deviation of values around the regression line
- the 95% credible set for the population mean of the Y variable for the subpopulation with `temp = 190`
- the 95% predictive interval for an individual observation of the Y variable when `temp = 190`

What quantities do we need to monitor?

Make the necessary adjustments to the dataset and N. Check the model, load the data (in two steps), compile for a single chain, and load initial values. Run 500 or 1000 iterations of burn-in. We will also use WinBUGS' rudimentary tools for checking how well models actually fit the data. Some of the plots in the "Compare" option on the "Inference" menu can be used to check the fit of simple linear regression models (one predictor) and for hierarchical linear models for which all the subjects have the same sequence of values of a single predictor variable. The WinBUGS manual calls this latter case "rectangular arrays of longitudinal data." Next week we will study models of this type.

To use plots of model fit you must monitor the *response variable* (in this case *pres*) and the parameter that represents its *expected value* (in this case  $\mu$ ) in addition to the parameters of interest.

Start monitoring and run 2000 iterations or so. (This regression model with centered predictor will converge virtually immediately.) Set "Use log" under the "Options" menu.

Then obtain history plots and autocorrelation plots for  $\alpha$ ,  $\beta$ ,  $\mu_{32}$ ,  $\sigma$ , and *pres*<sub>32</sub>.

Next, obtain summary statistics for the unknown quantities of interest.

- What is the 95% prediction interval for a new observation with *temp* = 190?
- What is the 95% credible set for the mean of *pres* values for the subpopulation in which *temp* = 190?

Select "Compare" from the "Inference" menu.

- Enter *pres* in the "other" box. This is the response variable.
- Enter *mu* in the "node" box.
- Enter *temp* in the "axis" box. This is the predictor variable.

Then click each of the four buttons at the bottom of the box.

- *scatterplot*  
This button produces a scatter plot of the observed values of the response variable versus the predictor variable. It enables you to see whether the relationship seems reasonably linear and whether there are any extreme outliers; you may decide that you need to check some data values, transform the response or predictor variable, or try something other than a straight-line model.
- *model fit* This plot displays the fitted regression line (based on posterior means of the intercept and slope), and the 95% central interval (dashed lines for 2.5 and 97.5 percentiles) for the *expected value* of the response variable corresponding to each value of the predictor.  
Why should you *not* expect 95% of the observed values to fall between the dashed lines?

### Transforming the response variable

Now let's try using the transformed response variable instead. Change the code as follows:

```
model
{
  temp.bar <- mean(temp[])
  for (i in 1:N) {
    logpres[i] ~ dnorm(mu[i], tau)
    mu[i] <- alpha + beta * (temp[i] - temp.bar)
    pres[i] ~ dnorm(0, .001) # have to account for unused vbl
  }
  sigma <- 1/sqrt(tau)
  alpha ~ dnorm(0, 1.0E-6)
  beta ~ dnorm(0, 1.0E-6)
  tau ~ dgamma(0.001, 0.001)
}
```

Then go through analogous steps as in the previous analysis.

### Looking ahead: Hierarchical normal linear model

If there is time, fit the "Rats" example from volume 1 of WinBUGS examples. This is an example of the type of model that we will be discussing in class next week. The data is on 30 baby rats who were weighed at 8, 15, 22, 29, and 36 days of age. The data values are their weights in grams.

Run a few burn-in iterations; then monitor  $Y$  and  $\mu$  and try the "Fit" operations. You will get separate plots for each subject for most types of plots.