9 Normal regression models

This is adapted from the book BMUW, chapter 5, plus some other stuff elsewhere.

9.1 Model specification

$$Y \mid X_1, \dots, X_p \sim \mathrm{N}(\mu(\beta, X_1, \dots, X_p), \sigma^2)$$

where

$$\mu(\beta, X_1, \dots, X_p) = \beta_0 + \beta_1 X_1 + \beta_p X_p = \beta_0 + \sum_{j=1}^{p} \beta_j X_j$$

The set of regression parameters: $\sigma^2, \beta = (\beta_0, \beta_1, \dots, \beta_p)$.

In BUGS code:

```
for(i in 1:n){
y[i] ~ dnorm(mu[i],tau)
mu[i] <- beta0 + beta1*X1[i]+ ... + betap*Xp[i]
}
s2 <- 1/tau  # for monitoring variance
s <- sqrt(s2) # for monitoring standard deviation
```

Independent prior distribution:

$$\beta_j \sim \mathrm{N}(\mu_{\beta_j}, c_j^2)$$

With $\mu_{\beta_j} = 0$ we have 'sceptical prior' (no effect of X_j is expected).

$$\tau \sim \Gamma(a, b)$$

so that σ^2 has inverse gamma prior with mean $b/(a-1)$ and variance $b^2/((a-1)^2(a-2))$. Note that we do not need to specify priors based on arguments of conjugate distributions. (In fact, for variance components (hierarchical models) it is even recommendable to try uniform priors for σ and then define `tau <- pow(sigma,-2)` to be used in normal model specification. Standard 'uninformative' conjugate priors may not be so uninformative as we might wish in some situations. This can be experimented with examples in hierarchical models).

To study the covariate effects, compute e.g. $P(\beta_j > 0 \mid y)$

```
p.betaj <- step(betaj)
```

or $\min(P(\beta_j > 0 \mid y), P(\beta_j < 0 \mid y))$. When zero value lies in the middle of the posterior, then this minimum would be close to 1/2. But if this minimum is below 2.5% (or whatever limit is of importance), then we conclude positive or negative effect.

Also might compute: $R_B^2 = 1 - \frac{\tau^{-1}}{s_Y^2} = 1 - \frac{\sigma^2}{s_Y^2}$, where s_Y^2 is sample variance of y. "the proportional reduction of uncertainty concerning the response Y achieved by incorporating the explanatory variables X".

```
sy2 <- pow(sd(y[]),2)
R2B <- 1 -s2/sy2;   s2 <- 1/tau
```

or: `R2B <- 1-1/(tau*sy2)` . If σ^2 is small, the model can predict the expected values of Y accurately and the inclusion of those X variables has been beneficial.

9.1.1 Ice cream consumption

Reference: Koteswara Rao Kadiyala (1970) Testing for the independence of regression disturbances. Econometrica, 38, 97-117. Also found in: Hand, D.J., et al. (1994) A Handbook of Small Data Sets, London: Chapman & Hall, 214.

Response variable: ice cream consumption in pints per capita over 30 four-week periods. Explanatory variable: mean temperature in Fahrenheit.

In this example, we could try both linear and nonlinear models. When comparing the models using DIC, we obtain negative DIC values, but this is not a problem. Only the difference in DIC matters. A probability density $\pi(y \mid \theta)$ can be greater than one, if the variance is small, leading to negative DIC. (See WinBUGS web site FAQ concerning more about DIC). If the difference is less than 5, then the models are nearly as good.

```
model{
for(i in 1:30){
Cons[i] ~ dnorm(mu[i],tau)
Cons2[i] ~ dnorm(nu[i],tau2)
Cons2[i] <- Cons[i]
mu[i] <- beta[1]+beta[2]*C[i]
nu[i] <- alpha[1]+alpha[2]*C[i]+alpha[3]*exp(C[i])
C[i] <- (F[i]-32)*5/9  # convert to Celsius
}
for(i in 1:2){beta[i]~dnorm(0,0.001)}
for(i in 1:3){alpha[i]~dnorm(0,0.001)}
tau ~ dgamma(0.001,0.001); s2 <- 1/tau
tau2 ~ dgamma(0.001,0.001); s22 <- 1/tau2
}
list(beta=c(1,0),tau=1,tau2=1,alpha=c(1,0,0))
```

```
 F[] Cons[]
  41 0.386
  56 0.374
  63 0.393
  68 0.425
  69 0.406
  65 0.344
  61 0.327
  47 0.288
```

```
   32 0.269
   24 0.256
   28 0.286
   26 0.298
   32 0.329
   40 0.318
   55 0.381
   63 0.381
   72 0.470
   72 0.443
   67 0.386
   60 0.342
   44 0.319
   40 0.307
   32 0.284
   27 0.326
   28 0.309
   33 0.359
   41 0.376
   52 0.416
   64 0.437
   71 0.548
END
```

9.1.2 York rainfall

The example is from P. Congdons book 'Bayesian Statistical Modeling', p.97. Explanatory variable x is the rainfall (in millimeters) in November, and the response y is that in December, in York, over 10 year period 1971-1980. The code below includes two versions, one with x variables directly, one with standardized x variables. The latter can be useful for better convergence in MCMC and less dependency in the joint posterior of model parameters (a, b). Also, the interpretation of the intercept a changes: it is either the expected December rainfall, given that November rainfall was zero (is that sensible?), or the expected rainfall, given an average rainfall in November.

```
 model{
 for (i in 1:n) {
 y[i] ~ dnorm(mu[i],tau.y)
 #mu[i] <- a + b*(x[i]-mean(x[])) # standardized x
 mu[i] <- a + b*x[i]
 }
 # priors
            a ~ dnorm(0,0.0001)
            b ~ dnorm(0,0.001)
            tau.y ~ dgamma(0.001,0.001)
            sig2.y <- 1/tau.y
 # prediction
  #mu.new <- a+b*(x.new-mean(x[])) # standardized x
  mu.new <- a+b*x.new
  y.new ~ dnorm(mu.new,tau.y)
}
 list(x=c(23.9,43.3,36.3,40.6,57,52.5,46.1,142,112.6,23.7),
 y=c(41,52,18.7,55,40,29.2,51,17.6,46.6,57),n=10,  x.new = 46.1)
```

9.1.3 Soft drink delivery times

This is from BMUW p. 157. Response variable: delivery time of an employee to refill soft drink vending machines (in minutes). Explanatory variables: (1) number of cases that need to be stocked in the machine, and (2) walking distance needed (measured in feet). The downloadable code, including data, of this model is available from the course page. Run the model and check results. You also get the posterior of the expected delivery time for a 'typical' situation. Note that parameter β_0 has no meaningful interpretation.

9.1.4 Atmospheric CO_2

Mauna Loa, Hawaii, atmospheric CO_2 records: the data consist of monthly mean CO_2 mole fraction determined from daily averages. The mole fraction of CO_2, expressed as parts per million (ppm) is the number of molecules of CO_2 in every one million molecules of dried air (water vapor removed).

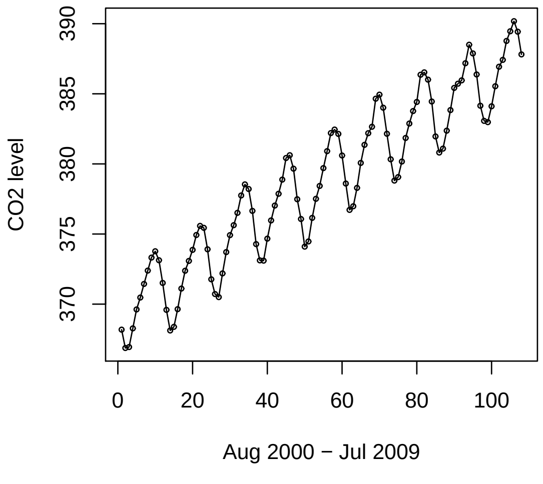

Figure 1: Part of CO_2 data.

The data show nearly linear increasing trend, with seasonal fluctuation. We can first fit a simple linear model, and then gradually refine that by adding nonlinear effects. The practical problem of inference can be to investigate whether there is evidence of nonlinearity. The practical problem of prediction can be to predict, or extrapolate, future (or past, or missing) observations based on observed data.

```
 model{
 tau ~ dgamma(0.01,0.01);
 for(i in 1:5){a[i] ~ dnorm(0,0.001)}
 for(i in 1:N){
 month[i] <- i
 x[i] ~ dnorm(mu[i],tau)
 mu[i]<- a[1]+a[2]*i+a[3]*sin(2*pi*i/12)+a[4]*cos(2*pi*i/12)
 }
 pi <- 3.1415926
 }
 # co2[-382]=co2[Sep1968]=320.41
list(N=120,x=c(368.18,366.87,366.94,368.27,369.62,370.47,
371.44,372.39,373.32,373.77,373.13,371.51,369.59,368.12,
368.38,369.64,371.11,372.38,373.08,373.87,374.93,375.58,
375.44,373.91,371.77,370.72,370.5,372.19,373.71,374.92,
375.63,376.51,377.75,378.54,378.21,376.65,374.28,373.12,
373.1,374.67,375.97,377.03,377.87,378.88,380.42,380.62,
379.66,377.48,376.07,374.1,374.47,376.15,377.51,378.43,
379.7,380.91,382.2,382.45,382.14,380.6,378.6,376.72,
376.98,378.29,380.07,381.36,382.19,382.65,384.65,
384.94,384.01,382.15,380.33,378.81,379.06,380.17,
381.85,382.88,383.77,384.42,386.36,386.53,386.01,
384.45,381.96,380.81,381.09,382.37,383.84,385.42,
385.72,385.96,387.18,388.5,387.88,386.38,384.15,
383.07,382.98,384.11,385.54,386.93,387.42,388.77,
389.46,390.18,389.43,387.81)
```

9.2 Vectors and multivariate priors

WinBUGS code can be made much more compact by using vector notations, although it does not necessarily make the model run any faster. The main advantage is clarity of model code, and easier use of multivariate priors instead of specifying univariate priors for every single parameter element. The mean of the normal model would then be coded as

```
 mu[i] <- beta0 + inprod(b[],x[i,])
```

And this can be made still more compact by defining the first row of x to contain ones, and including `beta0` as the first element in vector b:

```
 mu[i] <- inprod(b[],x[i,])
```

Then, if a normal-inverse gamma distribution is used for β, σ^2 we should have $\beta \mid \sigma^2 \sim \mathrm{N}_p(\mu, c^2 V \sigma^2)$ and $\sigma^2 \sim \mathrm{IG}(a, b)$, where p is the dimension of parameter vector β and c^2 is a parameter for the overall magnitude of prior variance. In Zellner's prior: $V = (X^T X)^{-1}$ with default choice $c^2 = n$. In BUGS code, the matrix $X^T X$ could be calculated within two loops where each of the elements becomes defined according to the matrix multiplication calculus. Then we can use the `inverse()`-command to calculate the matrix inverse. Hence, all that is needed is to define X in the data.

9.3 Cut function

Consider the following simple linear model:

$$y_i \sim \mathrm{N}(\beta_0 + \beta_1 x_i, \tau)$$

To generate example data in R, we could choose $x_i = 1, \ldots, 10$ and $\tau = 1/10^2$:

```
 x <- c(1,  5,  9, 13, 17, 21, 25, 29, 33, 37)
 y <- rnorm(10,x,10)
 plot(x,y)
```

10 data points were selected for the following BUGS model (the code includes two alternative models, concerning the x_i).

```
model{ for(i in 1:n){ # Lunn et al. 'The BUGS Project', SIM 2009.
  y[i] ~ dnorm(mu[i],tau)
  mu[i] <- inprod(beta[],X[i,])
  X[i,1] <-1
  X[i,2] <- x[i]  # exactly observed x
  #X[i,2] ~ dnorm(mux[i],0.1)# measurement error in x, without cut
  #X[i,2] <-cut(xx[i]);xx[i]~dnorm(mux[i],0.1)# measurement error, using cut
 mux[i] ~dnorm(x[i],0.01)# expected value of covariate x[i], "mx=x,px=1/100".
}
 for(i in 1:2){beta[i] ~ dnorm(0,0.001)}
 tau ~ dgamma(0.001,0.001)
}
 list(tau=1,beta=c(0,1)) # initial values
 list(n=10, x=c(1,  5,  9, 13, 17, 21, 25, 29, 33, 37),
 y=c(13.406485, 11.897630,  2.475133,  7.994367, 20.954012,
 24.909933, 16.610108, 38.865204, 31.991936, 32.861623))
```

If this model code is in the file `m2.bug`, then using R, the inputs need to be prepared, before running (alternatively, just run the above code directly in WinBUGS):

```
 n <- 10
 data <- list("x","y","n")
 parameters <- c("beta","tau","mu")
 inits<- list(list(beta=c(0,1),tau=1))
 res.sim <- bugs(data,inits,parameters,"m2.bug",n.chains=1,n.iter=2100,
                n.burnin=100,n.thin=1,program="openbugs")
 attach.bugs(res.sim)
 CI <- matrix(NA,10,3)
 for(i in 1:10){CI[i,1:3]<-quantile(probs=c(0.025,0.5,0.0975),mu[,i],names=FALSE)}
 plot(x,CI[,1],ylim=c(min(CI[,1]),max(CI[,3])))
 points(x,CI[,3])
 points(x,CI[,2],"l")
 points(x,y,col="blue",pch=16)
```

Now replace the model by the following (alternative model in the code above):

$$y_i \sim \mathrm{N}(\beta_0 + \beta_1 x_i^*, \tau) \quad x_i^* \sim \mathrm{N}(\mu_{x_i}, \tau_x) \quad \mu_{x_i} \sim \mathrm{N}(x_i, \tau_\mu)$$

by choosing a suitable fixed τ_x for the measurement error of the covariate, and τ_μ for the precision of the mean covariate value of the ith observation. The conditional mean of μ_{x_i} is chosen as the original x_i. (The model adapted from Lunn et al. 'The BUGS Project', SIM 2009).

In this case, the uncertain values x_i^* would also be estimated in the joint model of all uncertain quantities, drawing information from y_i which will influence the posterior estimates of x_i^*. Therefore, they are not *only* influenced by the model $\mathrm{N}(\mu_{x_i}, \tau_x)$. In the third model version, the uncertainty of x_i^* is described by the conditional distribution only, and we cut the bayesian inference by preventing learning from the y_i, concerning what we believe the x_i^* might be.

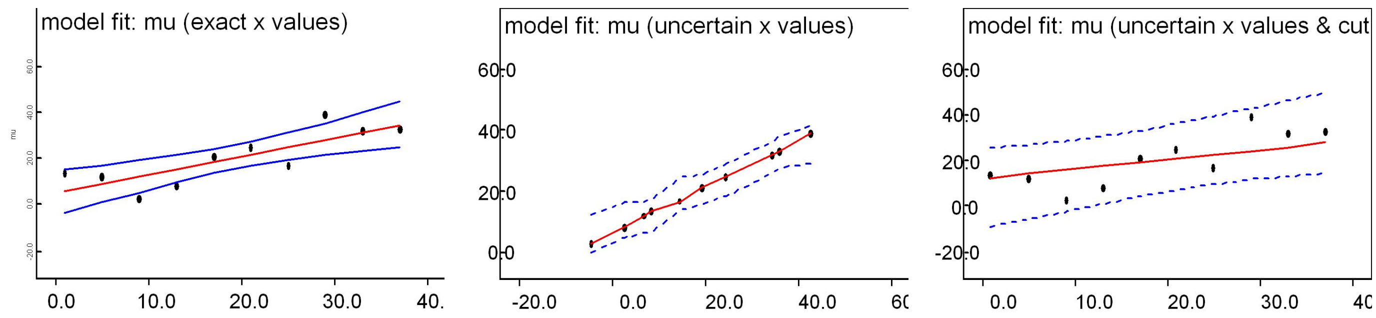

Figure 2: Results with (1) exact x_i, (2) uncertain x_i without cut, (3) uncertain x_i with cut-function.

An even simpler model below, showing same kind of behavior, because the values of x are highly uncertain. (We have very inaccurate information about x, based on the error model only).

```
 model{
 # simple measurement error in covariates, assuming large errors.
 # Therefore, much uncertainty about x-values.
 for(i in 1:n){
  y[i] ~ dnorm(mu[i],tau)
  mu[i] <- inprod(beta[],X[i,])
  X[i,1] <-1
 X[i,2] <- x[i]  # exactly observed x
 #X[i,2] ~ dnorm(x[i],0.01)  # measurement error in x, without cut
 #X[i,2]<-cut(xx[i]); xx[i]~dnorm(x[i],0.01)# measurement error, using cut
}
```

```
 for(i in 1:2){beta[i] ~ dnorm(0,0.001)}
 tau ~ dgamma(0.001,0.001)
}
```

But is the cut-function solving the problem or creating a new problem? Note that the probability calculus of the full model is destroyed by cut-function. Hence, the result does not represent the outcome of a fully bayesian analysis anymore. So what does it represent? There is some similarity with imputation methods, though. However, if the model without cut function produces unacceptable results, where to blame and what to correct? According to Box & Tiao (1973), in 'Bayesian Inference in Statistical Analysis':

Bayesian analysis automatically makes use of all the information from the data. It further follows that inferences that are unacceptable must come from inappropriate assumption and not from the inadequacies of the inferential system. Thus all parts of the model, including the prior distribution, are exposed to appropriate criticism.

Note that the cut-function is changing our inferential system, instead of looking at how our model could be improved!

However, the exact way of improving the model can be difficult too! Cut it out! But do we get imputation or amputation?

A presentation of cut-function by Nicky Best can be found from the IceBUGS-workshop:

`http://mathstat.helsinki.fi/openbugs/IceBUGS/IceBUGSAbstracts.html`

And further rationale for cutting is given in: Lunn et al: Combining MCMC with sequential PKPD modelling. J Pharmacokinet Pharmacodyn (2009) 36:19-38.