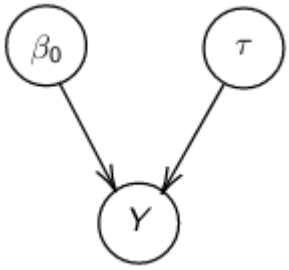
Bayesian Linear Regression

By now we know what a linear regression looks like. Let's consider a special case where number of parameters, p = 0:



Assume Y is distributed as a normal distribution with mean $_0$:
002

> = 1 /

$$P(y|\beta_0,\tau) = \sqrt{\frac{\tau}{2\pi}}e^{-\frac{\tau}{2}(y-\beta_0)^2}$$

Mean and variance:
E(Y) = $_0$; V(Y) = 1/$_0$ +

The Posterior Distribution for $_0$ is calculated using Bayes' Theorem

$$\beta_0|\tau, Y \sim N\left(\frac{\tau_0\mu_0 + \tau Y}{\tau_0 + \tau},\ \text{prec} = \tau_0 + \tau\right);$$

When Mean and Variance are Unknown

We use a Normal prior distribution for the mean 00

$$x \sim dgamma(r,\mu) = \frac{\mu^r x^{r-1} e^{-\mu x}}{\Gamma(r)}; x > 0$$

$$E(X) = \frac{r}{\mu}; V(X) = \frac{r}{\mu^2}$$

JAGS example:

```
model.1 <- "model {
for (i in 1:N) {
hbf[i] ~ dnorm(b.0,tau.t)
```

```
}
## prior on precision parameters
tau.t ~ dgamma(1,1);
### prior on mean given precision
mu.0 <- 5
tau.0 <- 0.44
b.0 ~ dnorm(mu.0, tau.0);


   ### prediction
hbf.new ~ dnorm(b.0,tau.t)
pred <- step(hbf.new-20) # hbf >= 20
}"
```

Predictive Distributions

Given the Prior and Observed data we can compute the probability of a new observation will be greater or less than some integer threshold. The predictive distribution is a distribution of unobserved y~, that is:

$$P(\tilde{y}|y_1,...,y_n) = \int_{-\infty}^{\infty} P(\tilde{y}|y_1,...,y_n,\beta_0,\tau)P(\beta_0,\tau|y_1,...,y_n)d\beta_0 d\tau$$

Since $P(\tilde{y}|y_1,...,y_n,\beta_0,\tau) = P(\tilde{Y}|\beta_0,\tau)$

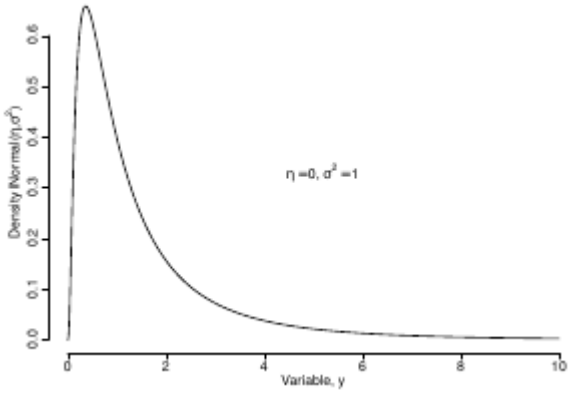
$$P(\tilde{y}|y_1,...,y_n) = \int_{-\infty}^{\infty} P(\tilde{y}|\beta_0,\tau)P(\beta_0,\tau|y_1,...,y_n)d\beta_0 d\tau$$

The two sources of variability in prediction are in the parameters V($_0$,

- 1.
  2.
-
-

Log Normal Distributions

Since in the example above the outcome distribution can only be positive we can use a Log-Normal distribution, a continuous distribution with support for values y > 0.



Y | η, σ^2 ~ lNormal(η, σ^2) with density function:

$$P(y|\eta,\sigma^2) = \frac{1}{y\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(\log(y)-\eta)^2}$$

Mean: $\mu = e^{\eta+\sigma^2/2}$
Median: e^{η}
Variance: $Var = \mu^2(e^{\sigma^2}-1)$

It also implies: Log(Y)~ Normal(η, σ^2)

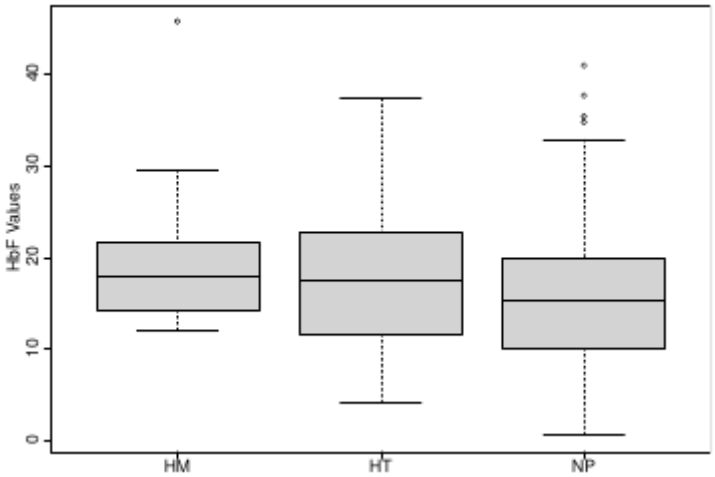
In JAGS we use: dlnorm(η, σ^2)

```
"model {
for (i in 1:N) {
hbf[i] ~ dlnorm(lb.0,tau.t)
}

## prior on precision parameters
tau.t ~ dgamma(1,1);

### prior on mean
lb.0 ~ dnorm(1.6,1.6);

b.0 <- exp(lb.0)

### prediction
hbf.new ~ dlnorm(lb.0,tau.t)
pred <- step(hbf.new-20)
}"
```

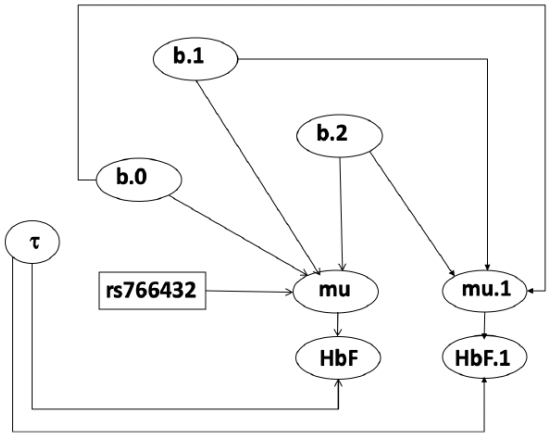
In the above example, we get the initial prior on $_0$ from previous data, where we derived median(b.0)=5 and V(b.0)=1/0.44=2.27
Then we take the log of the median to get η and transform the precision variable with: log(1 + V (b.0)/E (b.0)^2), then take the inverse of that to get the variance.

Parameter Interpretation

Let's consider data of the SNP rs766432 on the effect of HbF



Where HT is heterzygote, HM is homozygoute and NP is common alleles.



- We start by creating indicators for HT and NP using the equals(,) function in JAGS
  In the JAGS script

$$\text{equals}(e1, e2) = \begin{cases} 1 & e1 = e2 \\ 0 & \text{otherwise} \end{cases}$$

- The mean of Log HbF

$$\text{mu} = \begin{cases} \text{b.0} & \text{rs766432} = 1 \\ \text{b.0} + \text{b.1} & \text{rs766432} = 2 \\ \text{b.0} + \text{b.2} & \text{rs766432} = 3 \end{cases}$$

  b.1 = the effect of HT vs HM
  b.2 = the effect of NP vs HM
- The hypotheses:
  1. H0: b.1 = 0; HM and HT have the same effect
  2. H0: b.2 = 0; HM and NP have the same effect

```
model.1 <- "model {
for (i in 1:N) {
hbf[i] ~ dlnorm(mu[i],tau.t)
```

```
  mu[i] <- b.0+b.1 *equals(rs766432[i],2)+b.2 *equals(rs766432[i],3)
 }
 ## prior on precision parameters
 tau.t ~ dgamma(1,1);
 ### prior on mean given precision
 b.0 ~ dnorm(0, 0.001);
 b.1 ~ dnorm(0, 0.001);
 b.2 ~ dnorm(0, 0.001);
 ### prediction
 lmu.1 <- b.0;
  hbf.1 ~ dlnorm( lmu.1,tau.t);
    pred.1 <- step(hbf.1-20)
 lmu.2 <- b.0+b.1;
    hbf.2 ~ dlnorm( lmu.2,tau.t);
    pred.2 <- step(hbf.2-20)
 lmu.3 <- b.0+b.2;
    hbf.3 ~ dlnorm( lmu.3,tau.t);
    pred.3 <- step(hbf.3-20)
 ### fitted medians by genotypes
 mu.1 <- exp(lmu.1)
 mu.2 <- exp(lmu.2)
 mu.3 <- exp(lmu.3)
 par.b[1] <- b.0;
 qpar.b[2] <- b.1;
 par.b[3] <- b.2
 par.h[1] <- hbf.1;
    par.h[2] <- hbf.2;
    par.h[3] <- hbf.3;
 par.m[1] <- mu.1;
 par.m[2] <- mu.2;
 par.m[3] <- mu.3
 par.p[1] <- pred.1;
    par.p[2] <- pred.2;
    par.p[3] <- pred.3
}"

data.1 <- source("saudi.data.2.txt")[[1]]
model_hbf <- jags.model(textConnection(model.1), data = data.1,n.adapt = 1000)
update(model_hbf, 10000)
test_hbf <- coda.samples(model_hbf, c("par.b", "par.h", "par.m","par.p"), n.iter = 10000)
```

```
summary(test_hbf)
plot(test_hbf)
autocorr.plot(test_hbf)
```

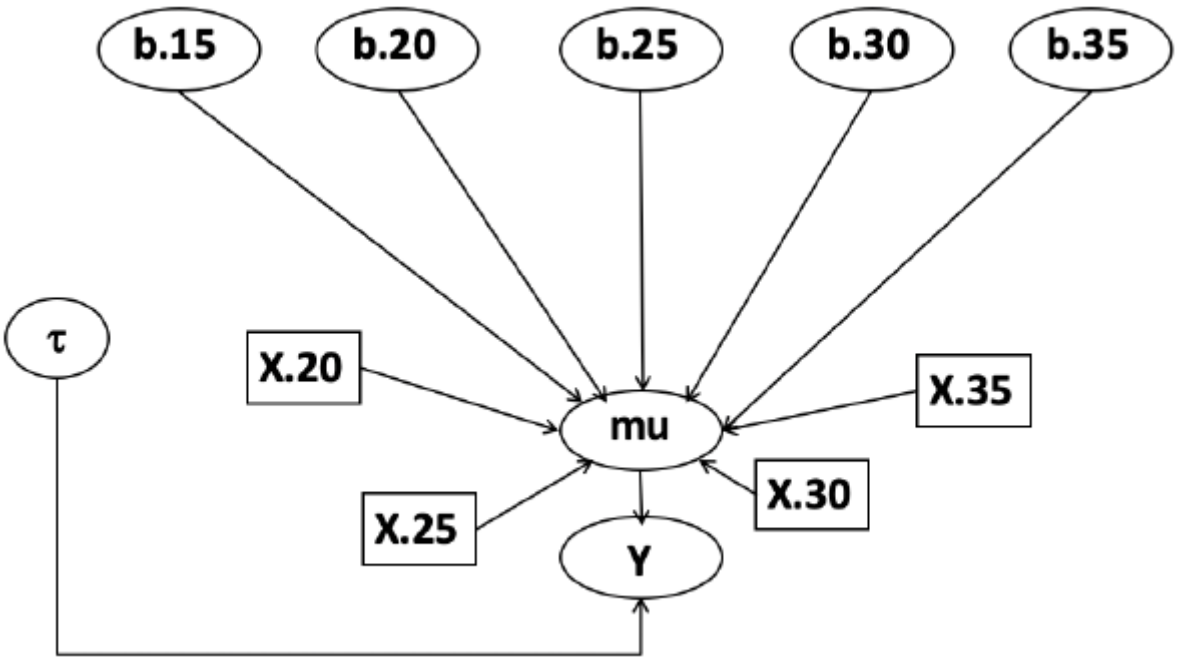
To analyze the convergence we can observe normality in auto-correlation plots. If we wee substantial auto-correlation (lag > X), we can repeat the the MCMC for 100,000 simulations and sample every X steps by using thin = X in coda.samples(); where X is how often order occurs in the plot.

test_hbf <- coda.samples(model_hbf, c("par.b", "par.h", "par.m","par.p"), n.iter = 1e+05, thin = 30)

Depending on which hypothesis we are testing we could also eliminate b.1 or b.2. Or in other situations reparameterizations can reduce correlation.

ANOVA Example

In this example let's consider a study of 5 different treatment groups assigned to wear shirts with differing levels of cotton (15%, 20%, 25%, 30%, and 35%) and strength was measured. We'll code these levels as dummy variables and



Note that since 15% is the reference group we keep it as a constant.

```
model.1 <- "model {
  ### data model
  for(i in 1:N){
    y[i] ~dnorm(mu[i], tau)
    mu[i] <- b.15 + b.20*lev.20[i] +b.25 *lev.25[i] +
```

```
  b.30*lev.30[i] +b.35 * lev.35[i]
    }
  ###
  prior
  b.15 ~dnorm(0,0.0001); ## referent group
  b.20 ~dnorm(0,0.0001);
  b.25 ~dnorm(0,0.0001);
  b.30 ~dnorm(0,0.0001);
  b.35 ~dnorm(0,0.0001);
  tau ~dgamma(1,1)
  ### difference in strength between level 3 (25%) and level 4 (30%)
  b.30.25 <- b.30-b.25
  ### estimated strength in groups (30%)
  strength[1] <- b.15
  strength[2] <- strength[1]+b.20
  strength[3] <- strength[1]+b.25
  strength[4] <- strength[1]+b.30
  strength[5] <- strength[1]+b.35
  }"
```

ANCOVA Example

These are models that include a continuous covariate and a categorical variable with 2 categories.

$$Y|\mu_i, \tau_i \sim N(\mu_i, \tau_i)$$

$$\mu_i = \beta_0 + \beta_1 x_{1i} + \alpha I_{2i} + \delta x_{1i} I_{2i}; \quad I_{2i} = \begin{cases} 1 & \text{if subject } i \text{ in group 2} \\ 0 & \text{otherwise} \end{cases}$$

When the slope differs in the 2 groups and the lines are not parallel

	Untreated	**vs.**	**Treated**	
Intercept:	β_0	$\rightarrow$	$\beta_0 + \beta_d$	shift by β_d
Slope	β_1	$\rightarrow$	$\beta_b + \beta_I$	shift by β_I

```
model.1 <- "model{
  ### data model
  for(i in 1:N){
     hbf_after[i] ~dlnorm(mu[i],tau)
     Lhbf_baseline[i] <- log(hbf_baseline[i])
     mu[i] <- beta.0 + beta.d*Drug[i] +
     beta.b*(Lhbf_baseline[i]-mean(Lhbf_baseline[])) +
     beta.i*Drug[i] *(Lhbf_baseline[i]-mean(Lhbf_baseline[]))
```

```
  }
  ### prior density
  beta.0 ~ dnorm(0,0.0001)
  beta.d ~dnorm(0, 0.0001)
  beta.b ~dnorm(0, 0.0001)
  beta.i ~dnorm(0,0.0001)
  tau ~ dgamma(1,1);
  ### inference
  parameter[1] <- beta.0
  parameter[2] <- beta.d
  parameter[3] <- beta.b
  parameter[4] <- beta.i
}"
### generate data
Drug = rep(0, nrow(hbf.data))
Drug[treatment == "Hy"] <- 1
table(Drug, hbf.data$Drug)
data.1 <- list(N = as.numeric(nrow(hbf.data)), hbf_baseline = hbf_baseline, hbf_after = hbf_after, Drug = Drug)
model_mean <- jags.model(textConnection(model.1), data = data.1,n.adapt = 1000)
update(model_mean, 10000)
test_mean <- coda.samples(model_mean, c("parameter"), n.iter = 10000)
```

So if the interaction term is significant then we would conclude the treatment has an effect

Missing Values

Treat missing values in the response as unknown parameters and JAGS will generate them as a form of imputation. Missing data in the covariates however is not so easy.

Model Selection: DIC

- Model selection based on marginal likelihood is most robust but difficult to implement
- Often model search over many models is based on BIC using posterior estimates of parameters
- Model selection for a small number of models is based on posterior intervals

Start from the deviance: -2log(P(y | β, τ))

Deviance information Criterion:
DIC = Dbar + pD = Dhat + 2 * pD

Dbar: -2 E(log(P(y | β, τ))) = posterior mean of the deviance
Dhat: -2 log(P(y | β^, τ^) = point estimate of the deviance using the posterior means of the

parameters
pD: Dbar - Dhat = Effective number of parameters

The model with the smallest DIC is estimated to be the model that would best predict a replicate dataset of the same structure as the observed.

Revision #4
Created 4 February 2023 00:12:21 by Elkip
Updated 5 February 2023 21:34:48 by Elkip