Lab 9: Multilevel Analysis in WinBUGS

Goals: 1. Use WinBUGS for multilevel modeling.
2. Learn the basic syntax of WinBUGS.
3. Learn the basics of using WinBUGS.

PART 0 Review Bayesian Analysis and WinBUGS

Consider the following conditional probability statement:

$$P(\theta \mid data) = \frac{P(data \mid \theta)\ P(\theta)}{P(data)},$$

where θ is the unobserved parameter that we want to learn about using the observed data.

P (θ | data) is the "posterior distribution" of θ. It describes the information on θ after combining prior knowledge P (θ) and what our data can inform us about θ through our statistical model, P (data | θ).

Since P(θ | data) is a probability distribution, statistical inference is made by examining the different characteristics of this distribution (eg mean, median, variance). WinBUGS draws samples repeatedly from the posterior distribution and we can calculate any statistics using these samples.

PART I A Two-stage Model: NMMAPS

Analysis goal: meta analysis of the relative risk estimates using two-stage Normal-Normal model.

<u>Data</u>:

Log Relative Risks for the six largest cities

City	Log RR estimate (% per 10 micrograms/m^3)	Statistical standard error	Statistical variance
Los Angeles	0.25	0.13	0.0169
New York	1.4	0.25	0.0625
Chicago	0.6	0.13	0.0169
Dallas/Ft worth	0.25	0.55	0.3025
Houston	0.45	0.40	0.1600
San Diego	1.0	0.45	0.2025

<u>Statistical Model</u>:

$$\log(RR_i) \sim N(\beta_i, \sigma_i^2)$$
$$\beta_i \sim N(\theta, \tau^2)$$

where RR_i = relative risk estimate for city *i*
β_i = true relative risk for city *i*
σ_i = estimated standard error for RR_i
θ = pooled RR
τ^2 = city-level heterogeneity for the true RR

Priors:

$$\theta \sim Normal\ (0, 1\times 10^6)$$

$$1/\tau^2 \sim Gamma\ (0.001, 0.001)$$

- We specify non-informative priors for the mean and variance similar to the example in Lab 8: a Normal prior with large variance for θ and a Gamma distribution for $1/\tau^2$.
- Recall that a Gamma distribution is non-negative and has "extremely" large standard deviation with the above parameters.
- We don't need to specify priors for β_i 's because we already put a Normal distribution assumption on them.

WinBUGS:

```
model{
      for( i in 1 : N) {
            log.rr[i] ~ dnorm(beta[i], prec2[i])
            beta[i] ~ dnorm (theta, prec.tau2)

            prec2[i] <-(1/se[i])*(1/se[i])
            }

      prec.tau2 ~dgamma(0.001,0.001)
      theta ~ dnorm(0.0,1.0E-6)


      tau <- sqrt(1 / prec.tau2)
      }
```

$\left.\right\}$ Statistical model

$\left.\right\}$ Priors

- We use "i" to index the cities (N = 6).
- Also note that WinBUGS takes "precision" instead of variance for the Normal distribution: precision = 1/variance. So we added the following two lines to tell WinBUGS to also track τ

```
p.sigma2[i] <-(1/se[i])*(1/se[i])
tau2 <- 1 / prec.tau2
```

Data:

```
list(N=6,   log.rr  = c(0.25,1.4,0.6,0.25,0.45,1.0),
            se =c(0.13,0.25,0.13,0.55,0.40,0.45))
```

Initial Values:

```
list(theta =0, prec.tau2=1)
```

Results (See Lab 8 p5 for detailed "clicking" instructions in WinBUGS):

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
theta	0.6448	0.2337	0.003771	0.2023	0.6377	1.123	2000	8001
tau	0.4309	0.253	0.006096	0.0951	0.3833	1.056	2000	8001
beta[1]	0.3074	0.1318	0.002303	0.05592	0.3053	0.5652	2000	8001
beta[2]	1.134	0.2714	0.007006	0.5944	1.136	1.656	2000	8001
beta[3]	0.6011	0.1236	0.001521	0.3619	0.6004	0.8508	2000	8001
beta[4]	0.4994	0.3576	0.004942	-0.2551	0.516	1.192	2000	8001
beta[5]	0.5476	0.2966	0.0039	-0.058	0.5496	1.126	2000	8001
beta[6]	0.7855	0.3251	0.005261	0.1897	0.7592	1.497	2000	8001

	WinBUGS	STATA
θ, overall log RR	0.64 (95% PI 0.20, 1.12)	0.66 (95% CI: 0.28, 1.04)
τ, between city SE	0.431	0.366

- Results from WinBUGS and Stata (meta package) are very simila.
- The mean of beta[i] in the above table represent the Bayes estimate of the true city-specific RR and the corresponding results from Stata are the empirical Bayes estimates. See Lab 4.

What if we change the prior distribution on θ to be more informative? Let's say

```
theta ~ dnorm(2,10)
```

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
theta	0.7538	0.1843	0.003397	0.4239	0.7417	1.146	2000	8001

Notice how the Bayesian estimate is pulled closer to 2, the prior mean.

Let's try a few other priors for θ to see the effects of prior distribution.

Prior mean	Prior Variance	Posterior mean	95% Posterior Interval
0	$1x10^6$	0.64	0.20 – 1.12
2	0.1	0.75	0.42 – 1.15
2	0.01	1.95	1.75 – 2.15
2	$1x10^{-6}$	2.00	1.998 – 2.002

PART II Profiling Health Care Providers

From the Winbugs help menu, open the "Institutional ranking" example (look under the help menu under Vol I examples).

Also see Lecture 8.

This example considers mortality rates in 12 hospitals performing cardiac surgery in babies. The data are shown below.

Hospital	No of ops	No of deaths
A	47	0
B	148	18
C	119	8
D	810	46
E	211	8
F	196	13
G	148	9
H	215	31
I	207	14
J	97	8
K	256	29
L	360	24

Statistical Model (Fixed-effect):

$$r_i \sim Binomial\ (\ p_i\ , n_i)$$

where r_i = No of deaths at hospital i and n_i = No of operations. Here we assume that the true failure probabilities (p_i) are *independent* (*i.e.* fixed effects) for each hospital. This is equivalent to assuming a standard non-informative prior distribution for the p_i's, namely:

Priors:

$$p_i \sim Beta\ (1, 1)$$

- The Beta distribution is a continuous distribution that only takes values between 0 to 1.
- We need a prior for *each* p_i.

WinBUGS:

```
model{
      for( i in 1 : N ) {
            r[i] ~ dbin(p[i], n[i]) #Model
            p[i] ~ dbeta(1.0, 1.0)  #Prior
            }
      }
```

Statistical Model (Random-effect):

$$r_i \sim Binomial\ (\ p_i\ , n_i)$$

$$\log\left(\frac{p_i}{1-p_i}\right) \sim Normal\ (\ \mu_i, \tau^2)$$

In this model, we assume that the logit of each hospital's rate is related to each other. Standard priors are given by:

$$\mu \sim Normal\ (0, 1\times 10^6)$$

$$1/\tau^2 \sim Gamma\ (0.001, 0.001)$$

Again we do not need to specify priors for p_i.

WinBUGS:

```
model{
      for( i in 1 : N ) {
            r[i] ~ dbin(p[i], n[i])        #Model
            b[i] ~ dnorm(mu, prec.tau2)
            logit(p[i]) <- b[i]
            }

      mu ~ dnorm(0.0,1.0E-6)               #Priors
      prec.tau2 ~dgamma(0.001,0.001)

      tau <- sqrt(1/prec.tau2)
      pop.mean <- exp(mu)/(1+exp(mu))
      }

#Data
list(n = c(47, 148, 119, 810, 211, 196, 148, 215, 207, 97, 256, 360),
     r  = c(0, 18, 8, 46, 8, 13, 9, 31, 14, 8, 29, 24), N = 12)
#Initial Values
list(p = c(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1),
mu = 0, prec.tau2 = 1)
```

Results:

For the fixed-effect analysis:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p[1]	0.01985	0.01936	2.167E-4	5.889E-4	0.01418	0.07133	2000	8001
p[2]	0.1267	0.02712	3.064E-4	0.07853	0.125	0.1845	2000	8001
p[3]	0.07431	0.02365	2.578E-4	0.03489	0.07192	0.1259	2000	8001
p[4]	0.05793	0.008254	8.127E-5	0.0427	0.05766	0.07498	2000	8001
p[5]	0.04242	0.01393	1.347E-4	0.01942	0.04079	0.07381	2000	8001
p[6]	0.07082	0.01815	2.179E-4	0.03949	0.06926	0.1102	2000	8001
p[7]	0.06688	0.02024	2.32E-4	0.03249	0.06484	0.1113	2000	8001
p[8]	0.1475	0.02379	2.625E-4	0.1044	0.1461	0.1982	2000	8001
p[9]	0.07212	0.01798	1.813E-4	0.04089	0.07071	0.1101	2000	8001
p[10]	0.09088	0.029	3.475E-4	0.04286	0.08809	0.1536	2000	8001
p[11]	0.1164	0.02013	2.177E-4	0.07978	0.1155	0.1589	2000	8001
p[12]	0.069	0.01353	1.279E-4	0.04518	0.06807	0.09808	2000	8001

For the random-effect analysis:

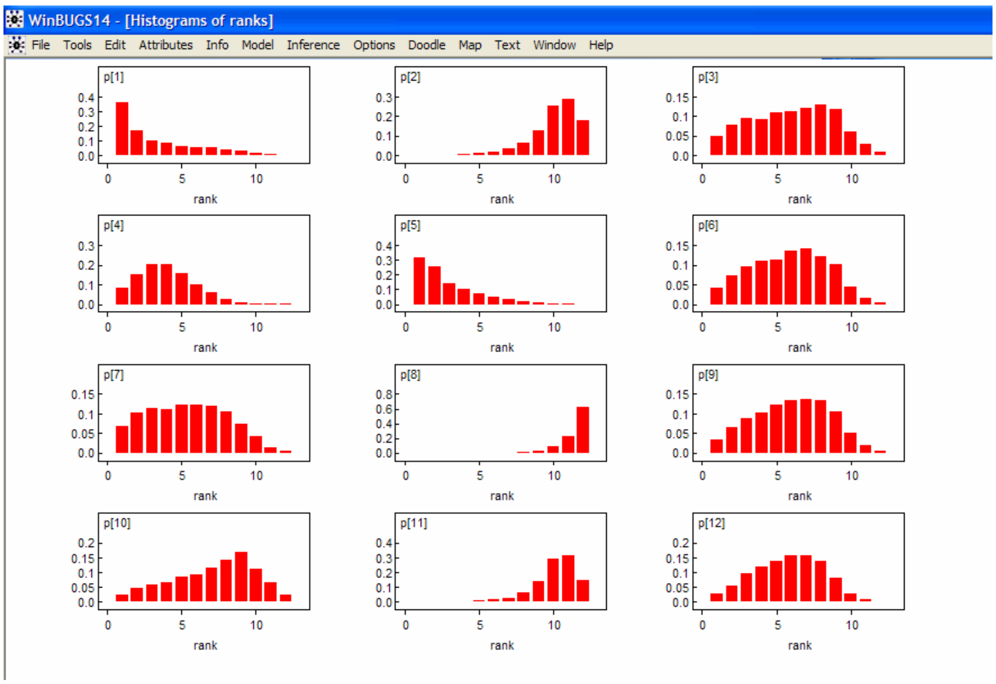
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
pop.mean	0.07261	0.01031	1.921E-4	0.05281	0.07243	0.0938	2000	8001
tau	0.405	0.1613	0.004142	0.1552	0.3813	0.7932	2000	8001
p[1]	0.05289	0.01953	3.809E-4	0.01804	0.05213	0.09348	2000	8001
p[2]	0.103	0.02198	3.46E-4	0.06724	0.1006	0.152	2000	8001
p[3]	0.07044	0.01731	2.224E-4	0.03954	0.06923	0.1082	2000	8001
p[4]	0.05928	0.007993	1.414E-4	0.04464	0.0589	0.07593	2000	8001
p[5]	0.0518	0.0133	2.692E-4	0.02781	0.05102	0.07944	2000	8001
p[6]	0.06904	0.01454	1.753E-4	0.0427	0.06855	0.1005	2000	8001
p[7]	0.06666	0.01599	2.186E-4	0.0382	0.06572	0.1005	2000	8001
p[8]	0.1229	0.02262	4.315E-4	0.08184	0.1221	0.1704	2000	8001
p[9]	0.06975	0.01436	1.593E-4	0.04419	0.06885	0.1004	2000	8001
p[10]	0.07847	0.01959	2.45E-4	0.04487	0.07678	0.1219	2000	8001
p[11]	0.1021	0.01765	2.8E-4	0.07157	0.1009	0.1398	2000	8001
p[12]	0.0685	0.01165	1.474E-4	0.04737	0.06793	0.09341	2000	8001

Hence using the random effects model, we estimate the population mean probability of failure as 0.0685 and standard deviation 0.405.

Ranking Hospital by Failure Rate

- One advantage of Bayesian analysis is the ability to estimate (particularly the uncertainty associated it) any function of the parameters by examining the corresponding posterior distribution.

- We can track the ranking of failure rate, 1 being the lowest, using the Rank tool (Menu → Inference → Rank).

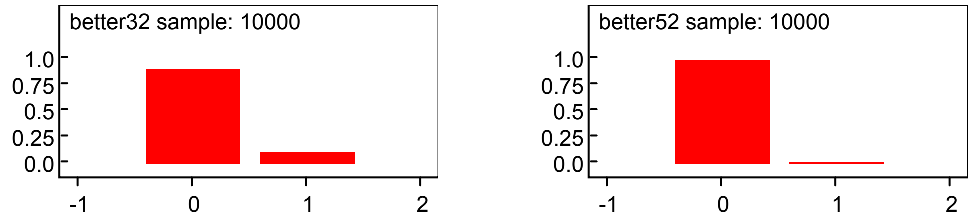


We see that there are quite a bit of uncertainty associated with the ranking for most hospitals. Only hospital 8 (H) and 11 (K) do not include the rank 6 in their 95% posterior interval.

- How about the ranking between two particular hospitals? Let's say we want to compare the failure rate between hospital 2 versus 5 and hospital 2 versus 3. This is done by simply including the following two lines in your "model" code.

```
better32 <- step(p[3]-p[2])
better52 <- step(p[5]-p[2])
```

The step function in WinBUGS will take value 1 if the value inside the parentheses is greater than zero. So the new variable "better32" will be 1 if at particular sampling draw p_3 is larger than p_2 and "better32" will be 0 otherwise. To make posterior inference, we will look at how often "better32" is 1.



node	mean	sd	MC error	2.5%	median	97.5%	start	sample
better32	0.1107	0.3138	0.003337	0.0	0.0	1.0	10001	10000
better52	0.0151	0.122	0.001648	0.0	0.0	0.0	10001	10000

So 11% of the time hospital 3 has a higher rate than hospital 2 and only 1.5% of the time hospital 5 has a higher rate then hospital 2.

You can fine more examples playing with posterior samples of the unknown parameters in Lecture 8.

PART III Random Intercepts for Growth Curve Model

Data: Weight gain in Asian children in Britain from Lab 6

Statistical Model:

$$weight_{ij} \mid age_{ij}, U_{1i} = (\beta_0 + U_{1i}) + \beta_1 age_{ij} + \beta_2 age_{ij}^2 + \varepsilon_{ij}$$
$$U_{1i} \sim N(0, \tau^2), \qquad \varepsilon_{ij} \sim N(0, \sigma^2)$$

We can re-rewrite this as:

$$weight_{ij} \mid age_{ij}, V_i = V_i + \beta_1 age_{ij} + \beta_2 age_{ij}^2 + \varepsilon_{ij}$$
$$V_i \sim N(\beta_0, \tau^2), \qquad \varepsilon_{ij} \sim N(0, \sigma^2)$$

where we assume that each individual's random intercept comes from a Normal distribution centered at the population average intercept. This is known as hierarchical centering and can improve estimation.

Re-code Data for WinBUGS (we will do this in Stata):

We have a total of 198 observations in 68 children:

(1) Create an index for each observation:
```
gen index =  _n
```
(2) Create an index for each child 1, 2, …, 68:
```
sort id
egen subject = group(id)
```
(3) Create the quadratic term for age:
```
gen age2 = age^2
```

So the dataset looks like:

id	occ	age	weight	brthwt	gender	**index**	**subject**	age2
45	1	.136893	5.171	4140	1	1	1	.0187396
45	2	.657084	10.86	4140	1	2	1	.4317596
45	3	1.21834	13.15	4140	1	3	1	1.484361
45	4	1.42916	13.2	4140	1	4	1	2.042493
45	5	2.27242	15.88	4140	1	5	1	5.163875
258	1	.19165	5.3	3155	2	6	2	.0367296
258	2	.687201	9.74	3155	2	7	2	.4722446
258	3	1.12799	9.98	3155	2	8	2	1.272372
258	4	2.30527	11.34	3155	2	9	2	5.314272

<u>WinBUGS Model</u>:

```
model{
      for (i in 1:198){
            mu[i] <- v[subject[i]] + beta[1]*age[i] + beta[2]*age2[i]
            weight[i] ~ dnorm(mu[i], prec.sigma2)
      }
      for (j in 1:68){
            v[j] ~ dnorm(beta0, prec.tau2)
      }

      beta[1] ~ dnorm(0.0,1.0E-6)
      beta[2] ~ dnorm(0.0,1.0E-6)
      beta0 ~ dnorm(0.0,1.0E-6)
      prec.sigma2 ~dgamma(0.001,0.001)
      prec.tau2 ~dgamma(0.001,0.001)

      tau <- sqrt(1/prec.tau2)
      sigma <- sqrt(1/prec.sigma2)
}
```

- We use i to index each row of observation. This corresponds to the variable *index*.
- We use j to index subject's random intercept. This corresponds to the variable *subject*.
- *mu* is the linear mean for the weight. Note that we use

  ```
  v[subject[i]]
  ```

  to tell WinBUGS to use the random intercept corresponding to subject[i] which can take values from 1 to 68.
- We use similar priors as in the previous examples.

<u>WinBUGS Data</u>:

Because the dataset is relative large this time, WinBUGS offers another way to input data. We first copy directly from Stata's data browser. Then we add square brackets following the variable name and put "END" at the last row. See the .odc file for detail. To load the data, highlight the first variable name.

WinBUGS Initial Values:

```
list( prec.sigma2 = 1, prec.tau2 = 1, beta0=1)
```

Results from xtmixed:

```
------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   7.817918   .2896529    26.99   0.000     7.250209    8.385627
        age2 |  -1.705599   .1085984   -15.71   0.000    -1.918448    -1.49275
       _cons |   3.432859   .1810702    18.96   0.000     3.077968     3.78775
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Identity                 |
                   sd(_cons) |   .9182256   .0973788      .7458965    1.130369
-----------------------------+------------------------------------------------
                sd(Residual) |   .7347063   .0452564      .6511507    .8289837
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =    78.07 Prob >= chibar2 = 0.0000
```

Results from MCMC:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta[1]	7.822	0.2942	0.01053	7.235	7.822	8.405	2000	8001
beta[2]	-1.707	0.1109	0.003511	-1.924	-1.707	-1.487	2000	8001
beta0	3.432	0.1829	0.005954	3.067	3.434	3.79	2000	8001
tau	0.9324	0.1016	0.001446	0.7521	0.9268	1.148	2000	8001
sigma	0.7449	0.04715	7.33E-4	0.659	0.7424	0.8457	2000	8001

PART IV Random Intercepts and Slopes for Growth Curve Model

We can easily include random slopes:

$$weight_{ij} \mid age_{ij}, U_{1i}, U_{2i} = U_{1i} + U_{2i} age_{ij} + \beta_3 age_{ij}^2 + \varepsilon_{ij}$$

$$\begin{pmatrix} U_{1i} \\ U_{2i} \end{pmatrix} \sim MVN\left(\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}, \Sigma = \begin{pmatrix} \tau_{11} & \tau_{12} \\ \tau_{21} & \tau_{22} \end{pmatrix} \right)$$

$$\varepsilon_{ij} \sim N(0, \sigma^2)$$

We will use the same prior distribution for β_1, β_2, β_1, and τ^2. For the covariance matrix we'll use a Wishart distribution (a multivariate scaled χ^2 distribution).

$$\Sigma^{-1} \sim Wishart\ (\Omega,\ p)$$

Ω = scale matrix, prior guess of order of covariance matrix.

p = degrees of freedom; to make vague should be equal to the number of random components. This has to same extremely-large variance effect like we previously picked Gamma prior for a single variance component.

WinBUGS Model (*okay so it's a bit more complicated*):

```
model{
   for (i in 1:198){
      mu[i] <- v[subject[i],1] + v[subject[i],2]*age[i] + beta3*age2[i]
      weight[i] ~ dnorm(mu[i], prec.sigma2)
   }
   for (j in 1:68){
      v[j,1:2] ~ dmnorm(beta[], prec.Sigma[,])
   }

   beta[1] ~ dnorm(0.0,1.0E-6)
   beta[2] ~ dnorm(0.0,1.0E-6)
   beta3 ~ dnorm(0.0,1.0E-6)
   prec.sigma2 ~ dgamma(0.001,0.001)
   prec.Sigma[1:2, 1:2] ~ dwish(Omega[,], 2)

   Sigma[1:2,1:2] <- inverse(prec.Sigma[,])
   sigma2 <- 1/prec.sigma2

   Omega[1,1] <- 1
   Omega[2,2] <- 1
   Omega[1,2] <- 0
   Omega[2,1] <- 0
}
#Initial Values
list(beta=c(1,1), beta3=1, prec.sigma2 = 1,  prec.Sigma =
structure(.Data=c(1,0,0,1),.Dim=c(2,2)) )
```

Priors

Values in Omega

GLLAMM Results

```
------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   7.703998     .24026    32.07   0.000     7.233097    8.174899
        age2 |  -1.660465   .0890109   -18.65   0.000    -1.834923   -1.486007
       _cons |   3.494512   .1376254    25.39   0.000     3.224771    3.764253
------------------------------------------------------------------------------
Variance at level 1
------------------------------------------------------------------------------
  .3315169 (.05826676)
Variances and covariances of random effects
------------------------------------------------------------------------------
***level 2 (id)

    var(1): .40444011 (.16452483)
    cov(2,1): .0880873 (.08802562) cor(2,1): .27478078

    var(2): .25409706 (.08865135)
------------------------------------------------------------------------------
```

MCMC Results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta[1]	3.512	0.1409	0.008069	3.226	3.513	3.78	2000	8001
beta[2]	7.66	0.2414	0.01953	7.21	7.659	8.153	2000	8001
beta3	-1.643	0.0905	0.007504	-1.831	-1.642	-1.476	2000	8001
sigma2	0.3353	0.05287	0.001487	0.241	0.3317	0.4518	2000	8001
Sigma[1,1]	0.4698	0.1532	0.005031	0.2255	0.4515	0.8203	2000	8001
Sigma[1,2]	0.05892	0.08045	0.002629	-0.1199	0.06615	0.1978	2000	8001
Sigma[2,1]	0.05892	0.08045	0.002629	-0.1199	0.06615	0.1978	2000	8001
Sigma[2,2]	0.3072	0.0897	0.002487	0.1704	0.2943	0.5161	2000	8001