Simple WinBUGS Programs for Common Situations

This section of the course will presents seven WinBUGS programs. Each program file contains three parts, the program listing, the data, and sample output using the program on the data. Most of the programs also contain an initial values section. Seven different types of simple analyses are presented:

1. Single binomial proportion
2. Single normal nmean, variance known
3. Single normal mean, variance unknown
4. Difference of two binomial proportions
5. Linear regression
6. Logistic regression
7. Hierarchical binomial proportion modelling

The online WinBUGS examples manuals (Volumes 1 and 2) contain 36 further types of analyses, almost all of them common to medicine. There is a very good chance that there will be a program already written for most analyses you will want to do (or something very close).

1. Single binomial proportion

Simple binomial model with x successes in n trials as the data, and a beta(1,1) prior density. Set up as a bernoulli model with success parameter θ, but also could have been set up as a binomial (see example 7).

Model

```
 model {
    for (i in 1:n)
      {
         x[i]   ~ dbern(theta);
       }
         theta ~ dbeta(1,1);  #  Prior density for theta
}
```

Data

list(x=c(0, 0, 0, 0 ,0, 0, 0 ,0 ,0 ,0 ,0 ,0 ,0 ,1, 1, 1, 1, 1, 1),
n=20)

Results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
theta	0.3177	0.0969	0.001411	0.1474	0.3116	0.5208	1001	5000

2. Single normal mean, variance known

Simple normal mean (μ) problem. The standard deviation of the data is sigma, which is obtained from the precision tau (value = 1). Prior are put on mu, note tiny prior precision, which implies huge variance, so close to a non-informative analysis. Data were randomly generated from a Normal(0,1) distribution. Note use of mean function.

Model

```
model {
    for (i in 1:n)
      {
         x[i]   ~   dnorm(mu,tau);
       }
         mu     ~   dnorm(0,0.0001);
         tau    <- 1;
         sigma  <-  1/sqrt(tau);
         xbar   <-   mean(x[]);
}
```

Data

list(x=c(-1.10635822, 0.56352639, -1.62101846, 0.06205707, 0.50183464,
0.45905694, -1.00045360, -0.58795638, 1.01602187, -0.26987089,
0.18354493 , 1.64605637, -0.96384666, 0.53842310, -1.11685831,
0.75908479 , 1.10442473 , -1.71124673, -0.42677894 , 0.68031412),
n=20)

Results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	-0.06355	0.2235	0.002986	-0.5034	-0.06203	0.3738	1001	5000

3. Single normal mean, variance unknown

Identical problem and data to the previous problem, but now variance is also considered unknown. A gamma(0.001, 0.001) prior is put on the precision, very widely dispersed (mean=1, variance = 1000), so close to noninformative. Note the use of an initialization file, nmeanvar.in. This is to avoid potential problems with bad starting values, which are likely in this problem due to very wide prior on the variance (through the precision).

Model

```
model {
    for (i in 1:n)
      {
         x[i]   ~ dnorm(mu,tau);
       }
         mu     ~ dnorm(0,0.0001);
         tau    ~ dgamma(0.001,0.001);
         sigma<-1/sqrt(tau);
         xbar <- mean(x[]);
}
```

Data

list(x=c(-1.10635822, 0.56352639, -1.62101846, 0.06205707, 0.50183464, 0.45905694, -1.00045360, -0.58795638, 1.01602187, -0.26987089 , 0.18354493 , 1.64605637, -0.96384666, 0.53842310, -1.11685831, 0.75908479 , 1.10442473 , -1.71124673, -0.42677894 , 0.68031412), n=20)

Initial Values

list(mu=1, tau=.33)

Results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	-0.06449	0.219	0.002937	-0.4952	-0.06569	0.3707	1001	5000
tau	1.088	0.3466	0.005788	0.5318	1.045	1.858	1001	5000
sigma	0.9971	0.1673	0.002776	0.7338	0.9781	1.372	1001	5000

Try changing some of the data points to be "missing" by changing the number to NA. For example, change the first data line

-1.10635822, 0.56352639, -1.62101846, 0.06205707, 0.50183464,

to

-1.10635822, 0.56352639, NA, 0.06205707, NA,

You will notice that WinBUGS has automatically detected the missing data, and performed a multiple imputation on its value. Note also that the standard deviation in the estimation of mu slightly increases, since only 18 values truly contribute independent information, rather than 20. Automatic imputation will work for any missing data that are on the left hand side of a WinBUGS statement. To impute other missing items, create a new line that describes its distribution. Note that this allows all data to be used in linear regressions, not only those cases with complete data.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	-0.00799	0.2329	0.002951	-0.4533	-0.01144	0.4728	1001	5000
sigma	0.9778	0.1817	0.002724	0.7053	0.9526	1.399	1001	5000
tau	1.148	0.395	0.005871	0.5108	1.102	2.011	1001	5000
x[3]	0.01365	1.018	0.01355	-2.018	0.003193	2.012	1001	5000
x[5]	-0.02405	1.034	0.01561	-2.092	-0.01312	1.927	1001	5000

4. Difference of two binomial proportions

Program to calculate the posterior distribution related to quantities comparing two independent binomial distributions. Posterior densities for the difference in proportions, risk ratio, and odds ratio are all calculated. Sample sizes in two groups need not be the same.

Model

```
model {
    for (i in 1:n1)
      {
         x[i]   ~ dbern(theta1);
       }
         theta1 ~ dbeta(1,1);
    for (i in 1:n2)
      {
         y[i]   ~ dbern(theta2);
       }
         theta2 ~ dbeta(1,1);

      propdiff <- theta1-theta2
      rr <- theta1/theta2
      or<- theta1*(1-theta2)/((1-theta1)*theta2)

}
```

Data

list(x = c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1),
n1 = 20,
y = c(0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1),
n2 = 25)

Results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
theta1	0.3176	0.09681	0.001334	0.148	0.3118	0.5211	1001	5000
theta2	0.7774	0.07969	0.001067	0.6057	0.7848	0.9103	1001	5000
propdiff	-0.4598	0.1254	0.001562	-0.6886	-0.4662	-0.1937	1001	5000
or	0.1505	0.112	0.001403	0.03093	0.122	0.4428	1001	5000
rr	0.4134	0.1358	0.001762	0.1862	0.4025	0.7155	1001	5000

5. Linear regression

Simple linear regression of dependent variable blood pressure (bp) on age and sex. Data were simulated from: bp = -25 + 2 * sex + 3 * age, so alpha = -25, beta1 = 2, beta = 3. Sigma was set to 5. Posterior means should be somewhat near these values, but with small sample size, do not expect high accuracy.

Model

```
model {
    for (i in 1:n) {
          mu[i] <- alpha + b.sex*sex[i] + b.age*age[i];
          bp[i]   ~ dnorm(mu[i],tau);
    }
    alpha    ~ dnorm(0.0,1.0E-4);
    b.sex     ~ dnorm(0.0,1.0E-4);
    b.age     ~ dnorm(0.0,1.0E-4);
    tau      ~ dgamma(1.0E-3,1.0E-3);
    sigma   <- 1.0/sqrt(tau);
 }
```

Data

list(sex = c(0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1),
age = c(59, 52, 37, 40, 67, 43, 61, 34, 51, 58, 54, 31, 49, 45, 66, 48, 41, 47, 53, 62, 60, 33, 44, 70, 56, 69, 35, 36, 68, 38),
bp = c(143, 132, 88, 98, 177, 102, 154, 83, 131, 150, 131, 69, 111, 114, 170, 117, 96, 116, 131, 158, 156, 75, 111, 184, 141, 182, 74, 87, 183, 89),
n=30)

Initial Values

list(alpha=50, b.sex=1, b.age=4, tau=1)

Results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	-23.0	3.615	0.2896	-30.05	-23.15	-15.23	1001	5000
b.age	2.934	0.06587	0.005214	2.792	2.937	3.065	1001	5000
b.sex	1.63	1.52	0.05885	-1.384	1.602	4.533	1001	5000
tau	0.06592	0.01799	3.943E-4	0.03531	0.06459	0.1043	1001	5000
sigma	4.01	0.5809	0.01299	3.097	3.935	5.322	1001	5000
mu[1]	150.1	1.032	0.02828	148.1	150.1	152.2	1001	5000
mu[2]	131.2	1.165	0.03362	128.9	131.2	133.5	1001	5000
mu[3]	85.57	1.418	0.09821	82.89	85.53	88.44	1001	5000

(etc...)

6. Logistic regression

Logistic regression model of a bone fracture with independent variables age and sex. The true model had: alpha = -25, b.sex = 0.5, b.age = 0.4. With so few data points and three parameters to estimate, do not expect posterior means/medians to equal the correct values exactly, but all would most likely be in the 95% intervals.

Model

```
model {
    for (i in 1:n) {
          logit(p[i]) <- alpha + b.sex*sex[i] + b.age*age[i];
          frac[i]   ~ dbern(p[i]);
    }
    alpha    ~ dnorm(0.0,1.0E-4);
    b.sex     ~ dnorm(0.0,1.0E-4);
    b.age     ~ dnorm(0.0,1.0E-4);
}
```

Data

list(sex=c(1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1),

age= c(69, 57, 61, 60, 69, 74, 63, 68, 64, 53, 60, 58, 79, 56, 53, 74, 56, 76, 72, 56, 66, 52, 77, 70, 69, 76, 72, 53, 69, 59, 73, 77, 55, 77, 68, 62, 56, 68, 70, 60, 65, 55, 64, 75, 60, 67, 61, 69, 75, 68, 72, 71, 54, 52, 54, 50, 75, 59, 65, 60, 60, 57, 51, 51, 63, 57, 80, 52, 65, 72, 80, 73, 76, 79, 66, 51, 76, 75, 66, 75, 78, 70, 67, 51, 70, 71, 71, 74, 74, 60, 58, 55, 61, 65, 52, 68, 75, 52, 53, 70),
frac=c(1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1),
n=100)

Initial Values

list(alpha=0, b.sex=1, b.age=1)

Results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	-22.55	5.013	0.6062	-34.33	-21.64	-14.29	1001	4000
b.age	0.3559	0.07771	0.009395	0.227	0.3418	0.5338	1001	4000
b.sex	1.405	0.7719	0.05094	-0.0387	1.374	3.031	1001	4000
p[1]	0.9575	0.03153	0.002943	0.879	0.9647	0.9952	1001	4000
p[2]	0.307	0.09828	0.004853	0.13	0.3012	0.5082	1001	4000
p[3]	0.6308	0.1041	0.003344	0.4166	0.6356	0.8178	1001	4000
p[4]	0.2477	0.103	0.007281	0.07738	0.2379	0.4728	1001	4000

(etc...)

7. Hierarchical binomial proportion modelling

Real data from the MSc thesis of Dr. Michael Schull. Nine MD's made decisions on 133 patients. The decisions involved whether to give the patient thrombolysis or not in the emergency room. There are Canadian guidelines as to whether to thrombolyse such patients or not, so each physician was evaluated as th whether they followed the guidelines for each subject they treated or not. For example, the first physician followed the guidelines in 19 out of 20 patients she/he treated. While each physician has their own rate of "success" (following the guidelines), it may be that overall, these rates may themselves have a distribution. The idea is to estimate both each physicians rate, as well as the rate of the "next" physician.

The latter is accomplished by looking at the posterior distribution of the "extra" variable "y". Note the hierarchical structure, the rates follow a distribution.

Model

```
model {
    for (i in 1:nmd) {
        x[i] ~ dbin(p[i],n[i]);

       logit(p[i]) <- z[i];
        z[i] ~ dnorm(mu,tau);
      }

           mu ~  dnorm(0,0.001);   # Prior distribution for mu
          tau ~ dgamma(0.001,0.001);   # Prior distribution for tau
          y   ~ dnorm(mu, tau) ;  #  Predictive distribution for rate
        sigma <- 1/sqrt(tau);      #  SD on the logit scale
            w <- exp(y)/(1+exp(y)); #  Predictive dist back on p-scale
}
```

Data

list(n=c(20, 6, 24, 13, 12, 4, 24, 12, 18),
x=c(19, 5, 22, 12, 11, 4, 23, 12, 16),
nmd=9)

Initial Values list(mu=0, tau=1)

Results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	2.642	0.3716	0.0294	1.973	2.63	3.357	1001	5000
tau	215.9	403.8	21.61	1.189	51.67	1447.0	1001	5000
sigma	0.2245	0.2467	0.01485	0.02631	0.1393	0.9187	1001	5000
p[1]	0.9301	0.02673	0.0018	0.8693	0.9332	0.9718	1001	5000
p[2]	0.9242	0.03447	0.002022	0.8463	0.9294	0.9689	1001	5000
p[3]	0.9277	0.0266	0.001763	0.8665	0.9316	0.9684	1001	5000
p[4]	0.9281	0.02856	0.001852	0.8598	0.9321	0.9712	1001	5000
p[5]	0.9279	0.0286	0.001851	0.8635	0.9319	0.9721	1001	5000
p[6]	0.9293	0.02987	0.001896	0.8636	0.9333	0.9758	1001	5000
p[7]	0.9313	0.02574	0.001803	0.8745	0.9343	0.9739	1001	5000
p[8]	0.932	0.02688	0.001838	0.8748	0.9347	0.9779	1001	5000
p[9]	0.9257	0.0288	0.001872	0.8595	0.93	0.9677	1001	5000
w	0.927	0.03535	0.001954	0.8524	0.9324	0.9725	1001	5000
y	2.638	0.4917	0.02993	1.753	2.624	3.565	1001	5000

Rerunning above, but with data changed so that first row is 10/20

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	2.347	0.6308	0.02193	1.315	2.286	3.806	1001	4000
p[1]	0.5969	0.1161	0.003224	0.366	0.6	0.8094	1001	4000
p[2]	0.8614	0.0981	0.001751	0.6063	0.8834	0.9825	1001	4000
p[3]	0.9076	0.05137	0.001297	0.7848	0.9152	0.9808	1001	4000
p[4]	0.9075	0.06156	0.001603	0.7596	0.9193	0.9889	1001	4000
p[5]	0.9027	0.06282	0.001518	0.7504	0.9153	0.9875	1001	4000
p[6]	0.9104	0.07954	0.001975	0.6975	0.9322	0.9972	1001	4000
p[7]	0.933	0.04373	0.001305	0.8296	0.9413	0.9918	1001	4000
p[8]	0.9404	0.05143	0.001714	0.8086	0.9537	0.9982	1001	4000
p[9]	0.8888	0.06163	0.001315	0.7372	0.8989	0.9775	1001	4000
sigma	1.205	0.5925	0.02313	0.366	1.106	2.64	1001	4000
tau	1.525	2.806	0.1283	0.1441	0.8176	7.476	1001	4000
w	0.8579	0.1527	0.002682	0.385	0.9027	0.9957	1001	4000
y	2.339	1.452	0.03215	-0.4683	2.228	5.446	1001	4000

Rerunning above, but with data changed so that first row is 100/200

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	2.399	0.6767	0.01764	1.236	2.345	3.935	1001	4000
p[1]	0.5088	0.03571	5.547E-4	0.4406	0.5092	0.5817	1001	4000
p[2]	0.8565	0.1035	0.002056	0.5942	0.8801	0.9838	1001	4000
p[3]	0.9104	0.05155	9.935E-4	0.7889	0.9198	0.9822	1001	4000
p[4]	0.91	0.06435	0.00122	0.746	0.9248	0.99	1001	4000
p[5]	0.9066	0.06611	0.001272	0.7396	0.9202	0.9895	1001	4000
p[6]	0.9191	0.08212	0.002	0.6984	0.9439	0.9985	1001	4000
p[7]	0.9397	0.04166	8.286E-4	0.8367	0.9485	0.9934	1001	4000
p[8]	0.9499	0.04702	0.001143	0.8217	0.9635	0.9989	1001	4000
p[9]	0.8894	0.06441	0.001169	0.7325	0.902	0.9784	1001	4000
sigma	1.433	0.6118	0.01963	0.6696	1.311	3.024	1001	4000
tau	0.7414	0.5728	0.01634	0.1097	0.5827	2.242	1001	4000
w	0.8457	0.1816	0.003351	0.2951	0.9117	0.9977	1001	4000
y	2.396	1.714	0.03377	-0.8707	2.335	6.091	1001	4000